

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema de programação por demonstração de um manipulador com garra

Ricardo Nuno Magalhães Borges

DISSERTAÇÃO

DISSERTAÇÃO

Orientador: Paulo José Cerqueira Gomes da Costa

Co-Orientador: Germano Manuel Correia dos Santos Veiga

19 de Julho de 2016

Resumo

Esta dissertação descreve uma solução de programação por demonstração baseada em métodos de observação de um manipulador com uma garra. Esta abordagem permite que o utilizador ensine ao sistema robótico as tarefas a realizar autonomamente evitando a necessidade de programar o manipulador segundo as ferramentas convencionalmente utilizadas.

Para conseguir isto recorreu-se à tecnologia de visão *3D*. Com esta tecnologia é possível seguir os movimentos realizados pelo utilizador, bem como extrair as coordenadas posição e orientação do movimento do mesmo. Enviando esses valores de coordenadas ao manipulador, este consegue posicionar a garra nessas coordenadas e desempenhar as funções que foram apresentadas.

De forma a permitir uma manobra de ensinamento mais realista, desenvolveu-se uma pequena garra impressa. Esta garra foi desenvolvida no software *Solidworks*, sendo mais tarde impressa com recurso a impressão 3D. O utilizador recorre a esta garra para fazer a demonstração de forma mais exata. Ao mesmo tempo esta garra possui marcadores de cor *RGB* que facilitam a deteção da mesma por parte da tecnologia *3D*.

A esta garra, foi adicionado um conjunto de sensores. Estes sensores fornecem dados relativos à força utilizada para agarrar um objeto. Também fornece dados sobre a orientação da garra e do ângulo de abertura da mesma. Assim, a esta garra de demonstração foi adicionado um integrado com um acelerómetro e giroscópio de três eixos. Na engrenagem será adicionado um potenciômetro capaz de medir o seu ângulo de abertura.

Todos estes dados são importantes para o ensinamento de um manipulador real, pois assim consegue-se saber com maior precisão a posição, a orientação e a força necessária a aplicar quando o objeto é agarrado. Sendo assim, foi criada uma interface gráfica em Lazarus capaz de receber e gravar toda esta informação. Gravar estas informações permite que mais tarde sejam usadas no manipulador que deve executar as tarefas requeridas.

Abstract

This thesis describes a way to program a gripper by demonstration, based on observation method. This approach allows the user to teach the robotic system how to carry out tasks autonomously, avoiding the need to program the manipulator conventionally.

To achieve this, it's necessary a 3D vision technology. With this technology it is possible to follow the user's movements and extract the initial and final coordinates of that movement. Then, these coordinate values will be sent to the gripper, allowing it to be positioned on the coordinates and perform the intended functions.

In order to make the most accurate measurements of the coordinates in the 3D space, a small gripper was developed in Solidworks and later printed using a 3D printing machine. At the same time the gripper incorporates a set of colored RGB markers that facilitate the 3D technology detection.

It was also added an instrumentation sensor set that provides information about the necessary force to grab an object, about the orientation of the gripper and its opening and closing angles. Thus, to this gripper was added an integrated circuit with an three-axis accelerometer and three-axis gyroscope. In order to measure its opening angle, a potentiometer will be incorporated in the gear of the gripper.

All these data are important to teach a real handler where to move an object from an initial position to a final position, more precisely. Thus it was created an interface in Lazarus able to receive and record all this information. Recording this information enables them to be later used in the gripper that must perform the required tasks.

Agradecimentos

Aqui estão os meus agradecimentos às pessoas que foram responsáveis pelo sucesso ao longo do percurso académico.

À minha família por todo o apoio ao longo destes anos e por terem sustentado o curso. Especialmente ao meu pai que durante parte do tempo em que estive a realizar o curso teve de lutar contra um cancro na coluna.

Ao meu orientador Prof. Paulo José Cerqueira Gomes da Costa pela disponibilidade que sempre demonstrou e por ter tido a paciência de me indicar a via correta ao longo de todo este projeto.

Aos técnicos Jorge Barbosa e Fernando Guedes, por disponibilizarem o material necessário à realização da dissertação e por todo o auxílio dado.

Gostaria de agradecer aos meus amigos fora da faculdade, especialmente a Catarina Coelho e Ricardo Ferreira por todo o apoio e auxílio estando sempre disponíveis a passar algumas horas na faculdade para ajudar. A Leonor Jalôto, sendo a pessoa que esta também a fazer a dissertação, é aquela que mais empatia demonstrou. A Susana “Nelas” Neves, que mesmo à distância, foi um pilar importante de apoio e de inspiração.

Um agradecimento especial a Carlos Leocádio, o homem das redes, Gonçalo Silva, do basket e João Silva, o soldador, tanto pelo auxílio como pelos momentos de descontração. A Cristina Rosinha, Sara Neves e Tânia Trindade por serem a minha família de faculdade. A Pedro Tavares, Hugo Costa e a Cláudia Rocha, por estarem presentes e ajudarem com boas trocas de ideias, durante a dissertação. A Diana Neves, que apesar de me pôr os cabelos brancos, foi uma boa parceira em muitas disciplinas, especialmente as de eletrónica. A Bruno Brito, por me dar uma visão bastante real do que será ser engenheiro.

A Alexandre Pires, por ser um bom parceiro de mesa, nos momentos em que mais se “partiu pedra”. A Duarte Ribeiro, por me deixar utilizar a sua mesa de trabalho e por fazer com que seja conhecido por “salvador de drones”. A Ivo Sousa, por ser a pessoa que ao fim do dia, diz sempre a frase que nos faz motivar.

A outros amigos de curso que estiveram presentes Bruno Martins, Rafael Piedade, Maria Ribeiro, Fátima Airosa, André Oliveira, Luís Gomes, Pedro Araújo, Hugo ribeiro, Gonçalo Correia, Tiago Silva, Ricardo Carvalho, Valter Costa, Vítor Morais, Tiago Cunha, Peter Cebola, Susana Pacheco Neves e Edgar Bento.

Agradecer ao café por ser a fonte de energia extra, apesar de me ter tornado um pouco mais dependente do mesmo.

Ricardo Nuno Magalhães Borges

Conteúdo

1	Introdução	1
1.1	Motivação	1
2	Revisão Bibliográfica	3
2.1	Tecnologias de reconhecimento de movimento	3
2.1.1	Wiimote	3
2.1.2	Kinect	4
2.1.3	Kinect 2	5
2.1.4	Asus xtion pro	5
2.1.5	Vantagens e desvantagens	6
2.2	Algoritmos de aprendizagem	6
2.2.1	Hidden Markov Model (HMM)	6
2.2.2	Neural Network (NN)	7
2.2.3	Support Vector Machine (SVM)	7
2.2.4	Self Organizing Map (SOM)	7
2.3	Hardware	8
2.3.1	Arduino	8
2.3.2	Acelerómetros	8
2.3.3	Sensores de força resistivos	9
2.3.4	Sensores de carga	9
3	Sistema de Visão	11
3.1	Algoritmo matemático	11
3.1.1	Distorção	13
3.2	Calibração	13
3.2.1	Condições de calibração	13
3.2.2	Software de calibração	16
3.2.3	Resultados da calibração	16
3.3	Medidas no mundo	19
4	Prototipagem da Garra	21
4.1	Estrutura da garra	21
4.1.1	Modificações realizadas no <i>Mini servo gripper Plate</i>	22
4.1.2	Novas peças desenvolvidas	25
4.1.3	Vantagens e desvantagens da impressão 3D	28
4.2	Sensores de força	29
4.2.1	Circuito equivalente	30
4.2.2	Vantagens do sensor	31

4.2.3	Fixação	31
4.3	Medição de ângulo	32
4.3.1	Medidas do potenciômetro	32
4.3.2	Fixação	33
4.4	Acelerômetro	34
4.4.1	Princípio de funcionamento	34
4.4.2	Comunicação	35
4.4.3	Vantagens da utilização do MPU6050	36
4.5	Marcadores de cor	36
4.5.1	Fixação	36
4.6	Plataforma de desenvolvimento	37
4.7	Interface Gráfica	38
4.7.1	Interface <i>ActiveKinect</i>	38
4.7.2	Interface <i>Calibration</i>	39
4.7.3	Interface <i>Gripper</i>	40
4.8	Conclusões	40
5	Resultados	43
5.1	Medidas de profundidade do Kinect	43
5.2	Coordenadas no mundo medidas	44
5.3	Aquisição e processamento de sinal	49
5.3.1	Calibração do HX711 para sensores de força	49
5.3.2	Calibração de valores do acelerômetro	51
5.4	Resultados das medidas da garra	53
5.5	Conclusões	63
6	Conclusão e Trabalho Futuro	65
6.1	Conclusões	65
6.2	Trabalho Futuro	65
A	Anexos	67
A.1	Ciruito Eagle do Hardware	67
A.2	Peças modificadas do <i>Mini servo gripper Plate</i>	69
A.3	Novas peças desenvolvidas	73

Lista de Figuras

2.1	Controlador Wiimote [1]	4
2.2	<i>Kinect</i> [2]	4
2.3	<i>Kinect 2</i> [3]	5
2.4	Asus Xtion PRO [4]	5
2.5	Exemplo de rede neuronal para reconhecimento gestos estáticos [5]	7
2.6	Arduino Uno [6]	8
2.7	Acelerómetro ADXL330 com 3 eixos	9
2.8	Sensor de força resistivo	9
2.9	Sensor de carga em forma de barra	10
2.10	Sensor de carga em forma de barra	10
3.1	Modelo pinhole [7]	12
3.2	Imagem original da câmara infravermelho	14
3.3	Imagem filtrada com objeto opaco da câmara infravermelho	14
3.4	Imagem do padrão de xadrez da câmara IR	15
3.5	Imagem do padrão de xadrez da câmara <i>RGB</i>	15
3.6	Calibração extrínseca da câmara IR	17
3.7	Calibração extrínseca da câmara <i>RGB</i>	18
3.8	<i>Kinect</i> com referências em ambas as câmaras [8]	19
4.1	Mini servo gripper Plate	21
4.2	Encaixe original	22
4.3	Encaixe para potenciômetro	22
4.4	Prato da garra impresso	23
4.5	Engrenagem direita do gripper original	23
4.6	Engrenagem direita do gripper modificada	24
4.7	Engrenagem após impressão	24
4.8	Dedo do Gripper original vs desenvolvido	25
4.9	Caixa do Hardware	26
4.10	Suporte da caixa do hardware	26
4.11	Encaixe para potenciômetro	27
4.12	Base do sensor de força	27
4.13	Barra de fixação do sensor de força	28
4.14	Peças impressas com sensores de força acoplados	28
4.15	Garra completa	29
4.16	Esquema elétrico do sensor de carga [9]	30
4.17	Esquema elétrico de dois sensores de carga ligados [9]	30
4.18	Integrado HX711 [10]	31

4.19	Estrutura de fixação dos sensores na garra	31
4.20	Estrutura com sensores fixados no dedo da garra	32
4.21	Relação entre medida do potenciômetro e medida em graus	33
4.22	Potenciômetro fixado na garra	33
4.23	Integrado do MPU6050 [11]	34
4.24	Eixos do MPU6050	35
4.25	Start e Stop da comunicação I2C [12]	35
4.26	Transmissão de informação do protocolo I2C [12]	36
4.27	Garra com marcadores <i>RGB</i>	37
4.28	Arduino nano	37
4.29	ActiveKinect	38
4.30	ActiveKinect	39
4.31	ActiveKinect	40
5.1	Calibração extrínseca da câmara <i>IR</i> , numa segunda posição	43
5.2	Calibração extrínseca da câmara <i>IR</i> , numa segunda posição	45
5.3	Calibração extrínseca da câmara <i>RGB</i> , numa segunda posição	46
5.4	Calibração extrínseca da câmara <i>IR</i> , numa terceira posição	47
5.5	Calibração extrínseca da câmara <i>RGB</i> , numa terceira posição	48
5.6	Força de medida pelos sensores	51
5.7	Ângulo de inclinação da garra	53
5.8	Posição da garra aberta	53
5.9	Posição da garra fechada	54
5.10	Agarrar o objeto	55
5.11	Preparação para agarrar objeto	55
5.12	Movendo o objeto	56
5.13	Objeto agarrado pelos sensores	56
5.14	Agarrar o objeto	58
5.15	Preparação para agarrar a pilha	58
5.16	Movendo o objeto	59
5.17	Pilha agarrada pelos sensores	59
5.18	Objeto em nova posição	61
5.19	Pilha agarrada pelos sensores	62
A.1	Circuito Eagle do Hardware	68

Lista de Tabelas

2.1	Vantagens e desvantagens de cada uma das tecnologias <i>RGBD</i> [13]	6
5.1	Primeiro teste de medidas do mundo real da câmara da <i>IR</i>	44
5.2	Primeiro teste de medidas do mundo real da câmara da <i>RGB</i>	44
5.3	Segundo teste de medidas do mundo real da <i>IR</i>	45
5.4	Segundo teste de medidas do mundo real da câmara da <i>RGB</i>	46
5.5	Terceiro teste de medidas do mundo real da câmara da infravermelho	48
5.6	Terceiro teste de medidas do mundo real da câmara da <i>RGB</i>	49
5.7	Amostra de valores de massa de dois objetos	50
5.8	Valores <i>RAW</i> do acelerómetro	52
5.9	Medidas adquiridas da garra	57
5.10	Medidas adquiridas de um segundo objeto	60
5.11	Medidas da garra com pilha AA em posição horizontal	63

Abreviaturas e Símbolos

NN	Neural Network
HMM	Hidden Markov Model
SVM	Support Vector Machine
SOM	Self Organizing Map
RGB	Red Green Blue
RGBD	Red Green Blue - Depth
IR	Infra Red
N	Newton
kg	Quilograma
SDA	Serial Data
SDL	Serial Clock
3D	3 Dimensões

Capítulo 1

Introdução

Nos últimos anos, a utilização de manipuladores robóticos nos processos de fabrico tem-se revelado imprescindível, uma vez que possibilita o aumento de eficiência dos mesmos. No entanto é necessário ter em conta que a reprogramação do procedimento de um equipamento robótico é de-veras complexa em ambiente industrial. Este fator implica o recurso a pessoal técnico qualificado, aquando da necessidade de alteração das tarefas a realizar numa linha de produção. De forma a superar estas dificuldades, a programação por demonstração surge como uma alternativa rápida e eficaz à reprogramação do equipamento robótico.

Com esta abordagem, que pretende facilitar a interação homem-máquina, a reprogramação de um equipamento é feita através da recolha de dados sobre os movimentos que o manipulador deve efetuar. Para tal é necessário recorrer a uma componente de visão robótica que deve ser conciliada com dados relativos a medições de *hardware* físico para o auxílio à captura de movimentos.

Na área da visão robótica, a tecnologia *RGBD* tem vindo a ganhar relevância em resultado da sua versatilidade e capacidade para extrair os dados necessários a este tipo de programação. O *Kinect*, comercializado pela *Microsoft*, é um exemplo de um equipamento com esta tecnologia o qual, através da captura de uma sequência de imagens, replica o movimento capturado num manipulador em tempo real ou guarda os dados obtidos por forma a poderem ser utilizados posteriormente. O processamento dos dados recolhidos através de uma câmara *RGBD* permite extrair informação relativa às coordenadas de um objeto no espaço *3D*. Contudo, existem fatores que permitem melhorar a replicação do movimento. Por isso, para o efeito, existe a necessidade de conciliar a visão *3D* com um conjunto de sensores, tendo em vista a extração das características mais importantes para o aumento da precisão deste movimentação, tais como orientação, abertura da garra e força necessária.

1.1 Motivação

O âmbito desta dissertação consiste no desenvolvimento de um sistema de aquisição de movimentos do braço de um operador humano, para que estes movimentos sejam, de seguida, processados num manipulador robótico. Os movimentos do operador serão adquiridos com recurso a uma

tecnologia *RGBD*, capaz de estimar movimento no espaço e que se caracteriza não só pela captação tradicional de imagem a cores, como também pela observação da distância a que os objetos se encontram.

Com isto, o operador, irá realizar os movimentos necessários para a execução da tarefa a ser programada. Estes movimentos realizados pelo operador representam as tarefas que o manipulador irá replicar. No entanto, a captura destes movimentos pode não ser a mais precisa. Assim, por forma a melhorar a precisão da captura dos seus movimentos, irá ser desenvolvida uma garra. Esta garra terá dois dedos e o seu formato irá de encontro às garras já existentes em certos modelos de manipuladores. Na garra serão colocados marcadores de cor, otimizando, assim, a aquisição do movimento, orientação e posição desta.

Os marcadores coloridos permitem localizar a garra de demonstração com maior precisão, por parte de câmara *RGB*. Este marcadores tem cor vermelho, verde e azul.

Para que o utilizador consiga perceber a orientação da garra irá recorrer-se a um integrado com acelerómetro e giroscópio de três eixos. Este equipamento permitirá determinar com mais exatidão a orientação da garra. Na engrenagem da garra será adicionado um componente capaz de medir o seu ângulo de abertura.

Para o processo de agarrar a peça, um aspeto a ter em conta é a força que pode ser aplicada. Para o efeito, a utilização de sensores de força na garra permitirá indicar o valor da força exercida. Com essa informação, será possível registar qual a força necessária para agarrar e deslocar a peça.

Capítulo 2

Revisão Bibliográfica

Neste capítulo serão exploradas as tecnologias de reconhecimento do movimento, nomeadamente algoritmos de aprendizagem que permitam aplicar programação por demonstração e interfaces gráficas para interação com o utilizador.

No âmbito das tecnologias de reconhecimento do movimento, o foco principal serão os sistemas utilizados das consolas de jogos tais como o *Wiimote* e o *Kinect*. Como já referido no capítulo anterior, um destes sistemas poderá ser usado para adquirir os movimentos do operador.

Os algoritmos de aprendizagem a ser explorados são os seguintes: *Hidden Markov Model*, *Neural Network*, *Support Vector Machine* e *Self Organizing Map*.

2.1 Tecnologias de reconhecimento de movimento

A deteção de movimento pode assentar numa combinação de sistemas de perceção da posição no espaço *3D*, como é o caso do *wiimote*. Uma outra forma de deteção do movimento poderá basear-se no *RGBD*, como é o caso do *Kinect* que permite visualizar o espaço em profundidade.

2.1.1 Wiimote

O *Wiimote* possui um acelerómetro com três eixos, um sensor de infravermelhos, um altifalante, um sistema de vibração e, para a comunicação, um sistema Bluetooth. Para detetar os movimentos realizados recorre à informação fornecida pelo acelerómetro ADXL330 e +/-3g 10% de erro.

Numa aplicação prática em manipuladores robóticos, o *Wiimote* indica ao robô os movimentos que este deverá realizar. Para tal, pressupõe-se a existência de uma informação prévia no manipulador acerca desses mesmos movimentos, que serão "ensinados" através de um algoritmo inteligente, a descrever mais adiante.

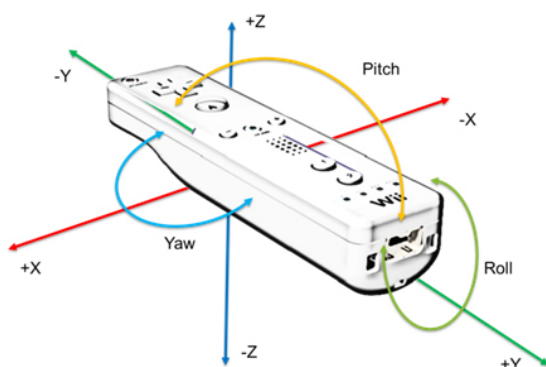


Figura 2.1: Controlador Wiimote [1]

2.1.2 Kinect

O *Kinect* é uma câmera que se baseia na tecnologia *RGBD*. Possui o mesmo sistema de cores que as câmaras normais, com o acréscimo de possuir um sensor de profundidade que permite observar a distância dos objetos à câmera. Assim, torna-se mais simples a detecção do movimento de um operador, pois este é capaz de registar o corpo inteiro ou partes do corpo, se estiver mais próximo do *Kinect* [14].

Figura 2.2: *Kinect* [2]

O sensor de profundidade do *Kinect*, possui um emissor de infravermelhos e um sensor CMOS monocromático sensível aos infravermelhos. Este compara a luz refletida com um padrão emitido, calculando assim a profundidade do que está a ser observado, com uma resolução de 640x480 e 30 FPS.

2.1.3 Kinect 2

O *Kinect 2* surge como uma forma melhorada do *Kinect 1*. A primeira grande melhoria esta na resolução da câmara de cor, que passa a ser de 1920x1080. A câmara de profundidade consegue medir a partir de 40 centímetros e já tem uma resolução de 512x424x16bpp. Nesta versão do *Kinect* o motor *tilt* que regula a inclinação do equipamento deixa de existir.



Figura 2.3: *Kinect 2* [3]

2.1.4 Asus xtion pro

Uma outra câmara com tecnologia *RGBD* existente é a Asus XtionPro Live, representada na Figura 4.11 que possui cinco características importantes: deteção de gestos, deteção de corpo inteiro, câmara *RGB*, câmara de profundidade e áudio [15]. A câmara de cor tem resolução 1280x1024 e a câmara de profundidade tem resolução 640x480 [16].



Figura 2.4: Asus Xtion PRO [4]

A deteção de gestos permite identificar, corretamente, os movimentos da mão do utilizador. A deteção do corpo inteiro é uma funcionalidade utilizada, maioritariamente, para vídeojogos. As colunas de áudio permitem a utilização deste equipamento em videoconferências [15].

2.1.5 Vantagens e desvantagens

A Tabela 2.1 apresenta as principais vantagens e desvantagens de cada uma das tecnologias *RGBD*. Do ponto de vista da utilização na área de desenvolvimento, é mais viável a utilização da primeira versão do *Kinect*, visto que as *drivers* são compatíveis com a maioria dos dispositivos e sistemas operativos.

Tabela 2.1: Vantagens e desvantagens de cada uma das tecnologias *RGBD* [13]

Equipamento	Vantagens	Desvantagens
Kinect	Drivers com boa qualidade Funcionamento estável com vários equipamentos diferentes Motor que regula a inclinação Bastante acessível no mercado	Menos compacto (24.9 x 6.6 x 6.7 cm) Necessário alimentação externa Pesado (1.36kg) Máximo 30fps
Asus Xtion Live	Mais leve (0.23kg) Mais pequeno (18 x 3.5 x 5 cm) Alimentação através da porta USB Qualidade da imagem <i>RGB</i> superior	Menos popular entre os utilizadores Menor qualidade das drivers Não tem motor para regular inclinação
Kinect 2	Qualidade de imagem HD Maior campo de visão	Dimensões (14 x 36 x 16.5 cm) Menor qualidade das drivers (principalmente em Linux) Não tem motor para regular inclinação

2.2 Algoritmos de aprendizagem

De forma a ser possível programar um manipulador por demonstração, tem sido aplicado um conjunto de algoritmos de aprendizagem, tais como *Hidden Markov Models*, *Neural Network*, *Support Vector Machine* ou *Self Organizing Maps*. Os referidos algoritmos têm sido utilizados para reconhecimento de gestos e classificação dos mesmos [5].

2.2.1 Hidden Markov Model (HMM)

O HMM é um algoritmo matemático, estocástico, de aprendizagem, baseado em probabilidades. Assenta num conjunto de estados finitos, onde o estado no instante atual t depende do estado anterior $t-1$. Define-se a probabilidade de transição aquela que caracteriza o estado atual J_t , dado o estado anterior I_{t-1} [17] [18].

Uma possível solução para aplicação do HMM em reconhecimento de gestos, baseia-se na classificação de gestos com base num conjunto de eventos discretos, cuja expressão é:

$$\lambda = \{A, B, \pi\} \quad (2.1)$$

onde A é a matriz de transição, B é a matriz de emissão e π é a distribuição de probabilidade inicial para cada estado [19].

2.2.2 Neural Network (NN)

Uma NN é um algoritmo de aprendizagem computacional que se baseia na forma como o sistema nervoso do cérebro processa informação. As NN têm mostrado ser uma método de identificação de gestos fiável, com boa capacidade de aprendizagem. No entanto dependendo da complexidade dos movimentos realizados e do processamento necessário, pode tornar-se difícil treinar uma NN [5].

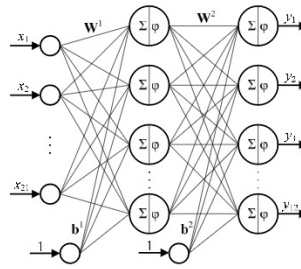


Figura 2.5: Exemplo de rede neuronal para reconhecimento gestos estáticos [5]

2.2.3 Support Vector Machine (SVM)

O algoritmo SVM é bastante popular entre os algoritmos de aprendizagem, por ser eficiente na extração de informação, mesmo quando se depara com a análise de amostras de grande dimensão ou que apresentam ruído [20]. Uma importante característica deste método é que consegue fazer a análise de amostras não lineares, não havendo assim necessidade de linearizá-las[21].

No reconhecimento de gestos este método é utilizado como forma de classificar gestos estáticos e dinâmicos, sendo por inerência um classificador binário. No entanto, pode acontecer que a classificação não esteja sempre correta. Por forma a superar isso, utiliza-se uma classificação multi-classe com SVM preparada para fazer o reconhecimento de gestos distintos[5].

2.2.4 Self Organizing Map (SOM)

O SOM é uma técnica de aprendizagem inteligente que não necessita de supervisão para ser ensinada. Tendo uma amostra bem definida, o SOM deverá ser capaz de identificar padrões comuns existentes na rede e seleccionar aqueles que lhe convêm. Este algoritmo é baseado em aprendizagem competitiva, onde as saídas competem entre elas para serem ativadas, resultando que apenas uma é ativada, a que se chama neurónio vencedor [22].

O SOM caracteriza-se por cinco fases:

1. *Initialization*, que escolhe valores aleatórios, entre os vetores iniciais.
2. *Sampling*, onde é definida a amostra de vetor de entrada a treinar.
3. *Matching*, onde é encontrado o neurónio vencedor mais próximo do vetor de entrada.

4. *Updating*, aplica a equação $\nabla w_{ji} = \eta(t) T_{j,I(x)}(t)(x_i - w_{ji})$.
5. *Continuation*, retorna ao passo 2, até que o esquema do mapa pare de mudar.

2.3 Hardware

Para melhorar o reconhecimento de gestos, deverá ser aplicado hardware capaz de auxiliar no reconhecimento dos mesmos, mas que seja, simultaneamente, capaz de fornecer informação extra ao manipulador e operador. Desta forma serão abordadas algumas tecnologias relativamente a plataformas de desenvolvimento, sensores e acelerómetros.

2.3.1 Arduino

O Arduino poderá ser uma uma plataforma de desenvolvimento, que permitirá a enviar e receber informação do hardware. Contém um micro-controlador e software para programar o mesmo, que usa uma versão adaptada de C++. É um equipamento barato e acessível de usar, que executa o programa desenvolvido sem necessitar de um sistema operativo próprio.

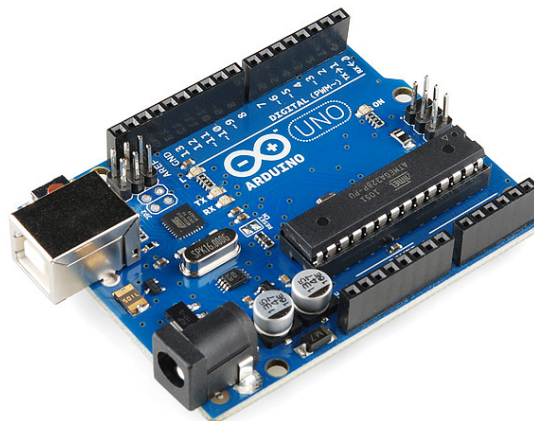


Figura 2.6: Arduino Uno [6]

2.3.2 Acelerómetros

Um acelerómetro serve para medir a aceleração de um objeto. Este dispositivo, ao adquirir a aceleração de um objeto, determina qual é o tipo de movimento do mesmo. Um acelerómetro pode ter mais do que um eixo, sendo esta particularidade é importante, pois com o aumento dos eixos podemos obter mais informação quanto à posição do objeto. Em particular no *wiimote* é usado um sensor ADXL330, de 3 eixos, onde podemos adquirir a aceleração nos eixos X, Y e Z [23].

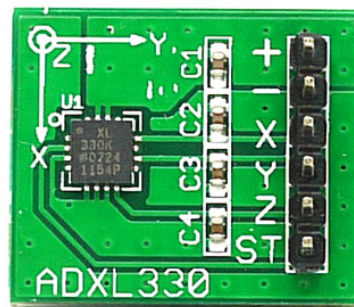


Figura 2.7: Acelerómetro ADXL330 com 3 eixos

2.3.3 Sensores de força resistivos

Os sensores de força resistivos têm uma superfície de contacto que, quando pressionada, mede o valor da força aplicada.



Figura 2.8: Sensor de força resistivo

O sensor de força da Figura 2.8, ao ser pressionado, faz variar o valor da resistência. Quanto maior for a força aplicada, menor será a resistência, com um alcance de medida entre 100g e 10Kg. Embora se consiga valores de medida interessantes, a precisão do mesmo não é a melhor [24]. Nas garras dos manipuladores este tipo de sensores são importantes, já que fornecem a informação da força aplicada quando a garra segura um objeto, percebendo-se se esta é adequada.

2.3.4 Sensores de carga

Uma outra forma de medir a força será utilizando sensores de carga. Os sensores de carga funcionam com o princípio da ponte de wheatstone. Estes sensores podem assumir formas físicas diferentes.

A Figura 2.9 mostra um sensor, da sparkfun [25], em forma de barra, que permite medir o peso de objetos colocados sobre o mesmo. Este sensor, por sua vez funciona de acordo com o princípio de uma ponte completa de wheatstone.



Figura 2.9: Sensor de carga em forma de barra

A Figura 5.6 mostra um sensor, da sparkfun [26], de forma quadrada, que permite medir o peso de objetos colocados sobre o mesmo, sendo utilizado, maioritariamente, em balanças das casas de banho. No caso deste sensor, ele funciona de acordo com o princípio de meia ponte de wheatstone.



Figura 2.10: Sensor de carga em forma de barra

Capítulo 3

Sistema de Visão

Este projeto tem como objetivo identificar uma garra, recorrendo a equipamento de visão $3D$. Neste capítulo proceder-se-á à descrição do algoritmo de calibração. Seguidamente serão calculados os parâmetros de calibração da câmara recorrendo à *toolbox* do *Matlab*. Por último extraem-se algumas conclusões sobre os resultados obtidos.

3.1 Algoritmo matemático

Aqui é descrito o processo matemático que permite, a partir de pontos da imagem $2D$ vista a partir da lente da câmara, que se determine os pontos do mundo real, ou seja, em $3D$. É importante realçar que também é possível efetuar uma transformação dos pontos no mundo, para pontos na imagem.

Num processo de calibração existem duas categorias de parâmetros diferentes:

- **Parâmetros intrínsecos:** são os parâmetros internos da câmara, que nos indicam as características da lente, tais como distância focal f , fatores de escala (α_x, α_y) , posição central dos pixels no plano ótico da lente (c_x e c_y) e distorção introduzida pelo sensor da câmara s ;
- **Parâmetros extrínsecos:** são características externas da câmara que, através da matriz de rotação R e do vetor de translação T , nos fornecem as coordenadas no mundo.

A calibração baseou-se no método de Heikkila[27] que consiste no modelo *pinhole*. Este modelo envia um feixe de luz em linha reta através de um pequeno orifício, no centro do plano da imagem.

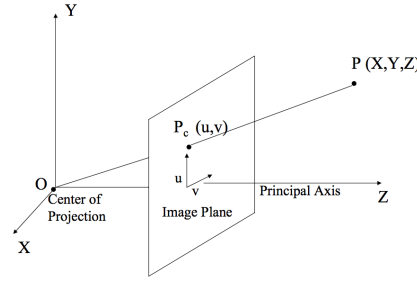


Figura 3.1: Modele pinhole [7]

Como demonstra a figura 5.11 podemos observar as coordenadas do modelo *pinhole*, onde a origem da projeção, O e a origem do plano da imagem são coincidentes. Desta forma, podemos relacionar as coordenadas dos dois referenciais através das seguintes equações:

$$\frac{u}{f} = \frac{x_c}{z_c} \quad (3.1)$$

$$\frac{v}{f} = \frac{y_c}{z_c} \quad (3.2)$$

Visto que a ótica da câmara normalmente não coincide com o sensor de imagem da mesma, as equações acima necessitam de sofrer uma transformação para poderem considerar esses *offsets*. Assim, será necessário adicionar uma translação c_x e c_y a cada eixo e, também considerar o mesmo para a distância focal, tendo assim f_x e f_y no eixo horizontal e vertical, respetivamente. Então, as novas equações são:

$$\frac{u}{f_x} = \frac{x_w}{z_w} + c_x \quad (3.3)$$

$$\frac{v}{f_y} = \frac{y_w}{z_w} + c_y \quad (3.4)$$

As equações acima representadas, relacionam as coordenadas 2D da câmara com as coordenadas 3D do mundo. Os parâmetros da câmara indicados correspondem, apenas, aos intrínsecos. É necessário, ainda, relacionar com os parâmetros extrínsecos. A equação seguinte mostra a influência dos parâmetros extrínsecos para determinar as coordenadas no mundo.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} R|T \end{bmatrix} * \begin{bmatrix} W_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M * \begin{bmatrix} R|T \end{bmatrix} * \begin{bmatrix} W_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.5)$$

Na equação 3.5, M é a matriz dos parâmetros intrínsecos, R a matriz de rotação e T o vetor de translação. As variáveis X_w , Y_w e Z_w representam as coordenadas no espaço 3D.

3.1.1 Distorção

Pode-se considerar dois tipos de distorção: a radial e a tangencial. A distorção radial, representa a variação da posição em relação ao raio da lente e causa na imagem um efeito que toma a forma de "barril". Por sua vez, a distorção tangencial representa o desvio de posição entre o plano da imagem e o plano da lente. Tipicamente, esta distorção ocorre porque a lente não se encontra alinhada com o plano da imagem.

A equação 3.5 não tem em conta a a distorção da câmara. Para solucionar esta questão, considerou-se o modelo apresentado por Brown [28]. Assim é necessário alterar as equações 3.3 e 3.4.

$$u = x_d * f_x + c_x \Leftrightarrow x_d = \frac{u - c_x}{f_x} \quad (3.6)$$

$$v = y_d * f_y + c_y \Leftrightarrow y_d = \frac{v - c_y}{f_y} \quad (3.7)$$

Nas equações 3.6 e 3.7 as variáveis x_d e y_d são as variáveis de distorção da lente e tomam a seguinte forma:

$$x_c = (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) * x_d + 2 * p_1 * x_d * y_d + p_2 * (r^2 + 2 * x_d^2) \quad (3.8)$$

$$y_c = (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) * y_d + p_1 * (r^2 + 2 * y_d^2) + 2 * p_2 * x_d * y_d \quad (3.9)$$

Nas equações 3.8 e 3.9 as variáveis x_d e y_d são as coordenadas distorcidas, x_c e y_c são as coordenadas não distorcidas. A variáveis k_1 , k_2 e k_3 são os coeficientes de distorção radial enquanto que as variáveis p_1 e p_2 representam os coeficientes de distorção tangencial. Já a variável r^2 é determinada da seguinte forma:

$$r^2 = x_d^2 + y_d^2 \quad (3.10)$$

3.2 Calibração

A calibração de uma câmara permite adquirir os parâmetros já acima descritos. Estes são utilizados para relacionar as coordenadas $2D$, vistas a partir da câmara, com as coordenadas $3D$ no mundo. Assim são determinados os parâmetros intrínsecos, que correspondem às características internas da câmara e que uma vez determinados, não dependem da posição da mesma. Os parâmetros extrínsecos relacionam a posição da câmara com um referencial no mundo [29].

3.2.1 Condições de calibração

Para levar a cabo a calibração utiliza-se um padrão de xadrez 9 por 6, impresso numa folha A3, onde cada quadrado tem 37 milímetros de lado. O padrão de xadrez foi colocado em várias

posições de forma a obter um conjunto alargado de imagens que permitam uma boa calibração. O número de fotografias tiradas foi um total de 50: 25 para a câmara de infravermelhos e 25 para a câmara *RGB*. Foi necessário obstruir o emissor de infravermelhos do *Kinect*, com um objeto de material opaco, de forma a filtrar as irregularidades criadas pelos feixes projetados. As Figuras 3.2 e 3.3 mostram a diferença entre a imagem original sem filtro e a imagem com filtro.

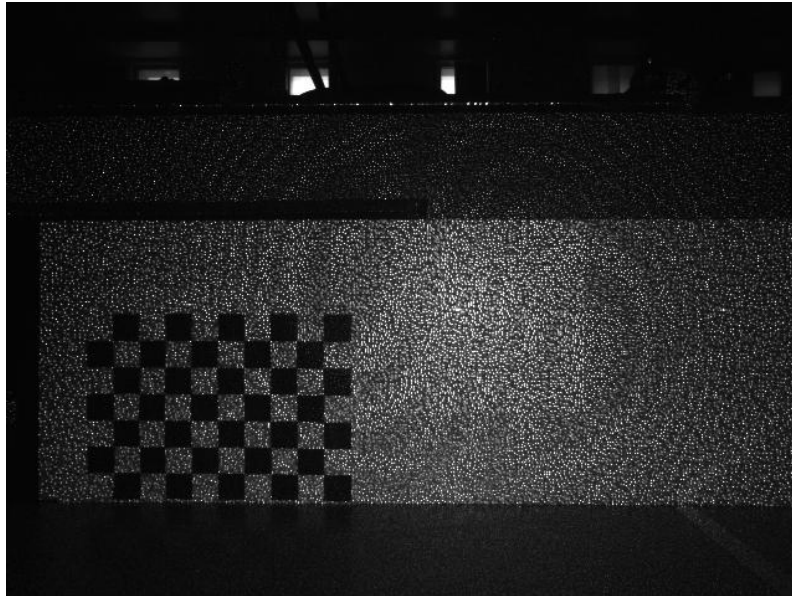


Figura 3.2: Imagem original da câmara infravermelho

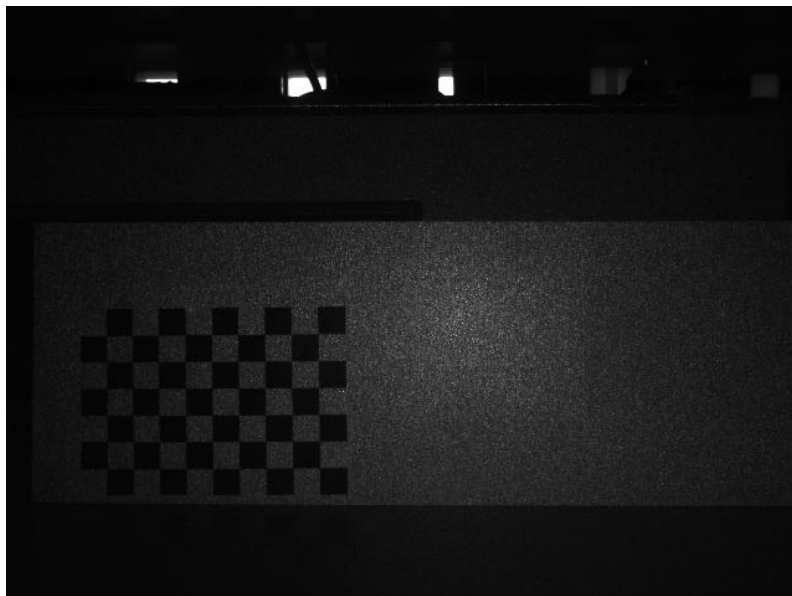


Figura 3.3: Imagem filtrada com objeto opaco da câmara infravermelho

Foram tiradas fotografias do padrão de xadrez, com o *Kinect*, às distâncias de 70, 90 e 110

centímetros. Em cada uma destas distâncias, o padrão esteve em posição frontal e centrada e foi, também, rodado 15 e 30 graus no sentido dos ponteiros do relógio e no sentido anti-horário. À distância de 110 centímetros, o padrão foi colocado em posição frontal e rodado 15 graus, mas em vez de estar centrado, foi posicionado lateralmente. No que se refere às restantes distâncias o padrão foi colocado frontalmente, voltado para a câmara, mas posicionado lateralmente.

As Figuras 3.4 e 3.5, são dois exemplos representativos das imagens infravermelho e *RGB*, que foram fotografadas para a calibração.

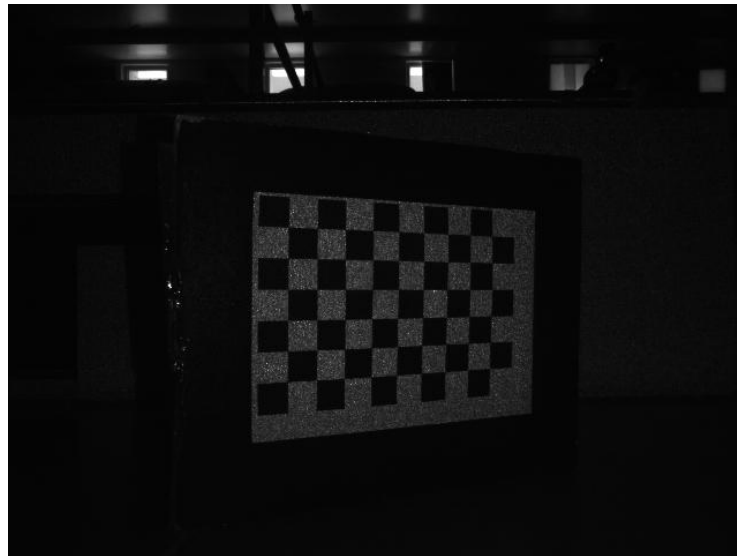


Figura 3.4: Imagem do padrão de xadrez da câmara IR

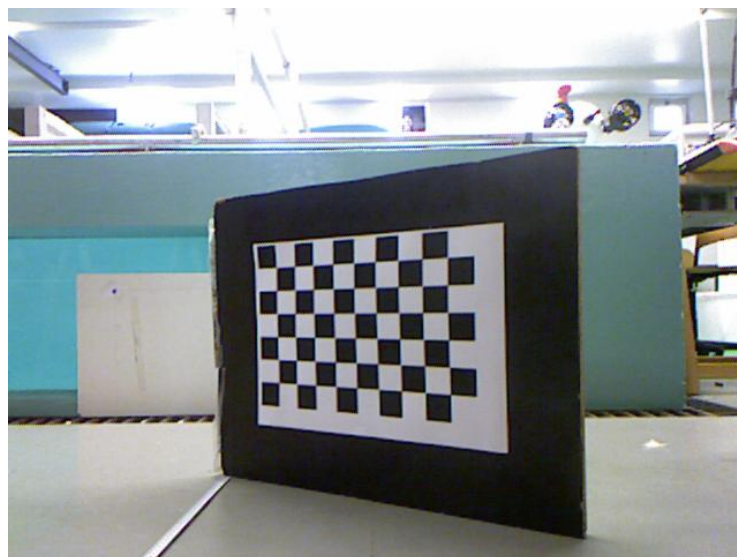


Figura 3.5: Imagem do padrão de xadrez da câmara *RGB*

Um outro aspeto que condiciona uma boa calibração, é a luz ambiente. De modo a obter menor

interferência, as fotografias do padrão de xadrez, foram tiradas num ambiente de *indoor*, com luz controlada e com menor incidência de luz natural possível.

3.2.2 Software de calibração

O software escolhido para a calibração da câmara foi o Matlab, com a *Camera Calibration Toolbox*, desenvolvida por Jean-Yves Bouguet [30]. Esta *Toolbox* tem a vantagem de estar a ser executada num software matemático bastante evoluído e do ponto de vista da utilização, dá uma maior liberdade ao utilizador na escolha da origem dos referenciais, no padrão de xadrez.

No caso da calibração extrínseca, escolher a origem do referencial torna-se importante, pois assim consegue-se escolher o referencial que mais facilita as medidas no mundo real.

3.2.3 Resultados da calibração

Na calibração, o *Kinect* manteve-se fixo a uma altura do solo de 28 centímetros. Os parâmetros intrínsecos, distorção D e Matriz de câmara A , tomam o seguinte formato:

$$D = \begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \\ k_3 \end{bmatrix}$$

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Na matriz D encontram-se os parâmetros da distorção radial k e tangencial p . A matriz A contém as distâncias focais f em milímetros e o centro ótico c são as coordenadas em pixel, sendo estes parâmetros constantes, até que se mude alguma característica interna da câmara, como por exemplo uma lente. Após a realização do processo de calibração das duas câmaras, extraíram-se os seguintes valores dos parâmetros intrínsecos.

$$D_{IR} = \begin{bmatrix} -0.1407 \\ 0.2658 \\ -0.0082 \\ 0.0016 \\ 0 \end{bmatrix}, \quad D_{RGB} = \begin{bmatrix} 0.1477 \\ -0.3307 \\ 0.0049 \\ 0.0019 \\ 0 \end{bmatrix}$$

$$M_{IR} = \begin{bmatrix} 572.10 & 0 & 328.37 \\ 0 & 569.82 & 240.47 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_{RGB} = \begin{bmatrix} 508.79 & 0 & 310.44 \\ 0 & 506.21 & 259.40 \\ 0 & 0 & 1 \end{bmatrix}$$

Observando os resultados constata-se que a distância focal da câmara de infravermelhos é superior à de cor. Quanto ao centro ótico, o da câmara infravermelhos está, aproximadamente, nas coordenadas (328,240) e o da câmara *RGB* está em (310, 259). Sendo que o *Kinect* tem uma resolução de 640 x 480, teoricamente o centro seria em (320, 240). Assim, pode concluir-se que o centro ótico da câmara *RGB* se encontra bastante desviado daquilo que seria o seu valor teórico, enquanto que a câmara de infravermelhos, apenas evidencia um ligeiro desvio no eixo *v*.

Os parâmetros extrínsecos aqui demonstrados, são referentes às imagens das Figuras 3.6 e 3.7. Nestas figuras a câmara do *Kinect* encontrava-se a um metro de distância do padrão de xadrez. A origem escolhida do padrão de xadrez tinha uma translação à esquerda de 45 centímetros em relação ao centro do *Kinect*. Este estava posicionado sobre uma superfície plana, a 28 centímetros do solo. Os parâmetros aqui representados correspondem às matrizes de rotação e aos vetores de translação de ambas as câmaras.

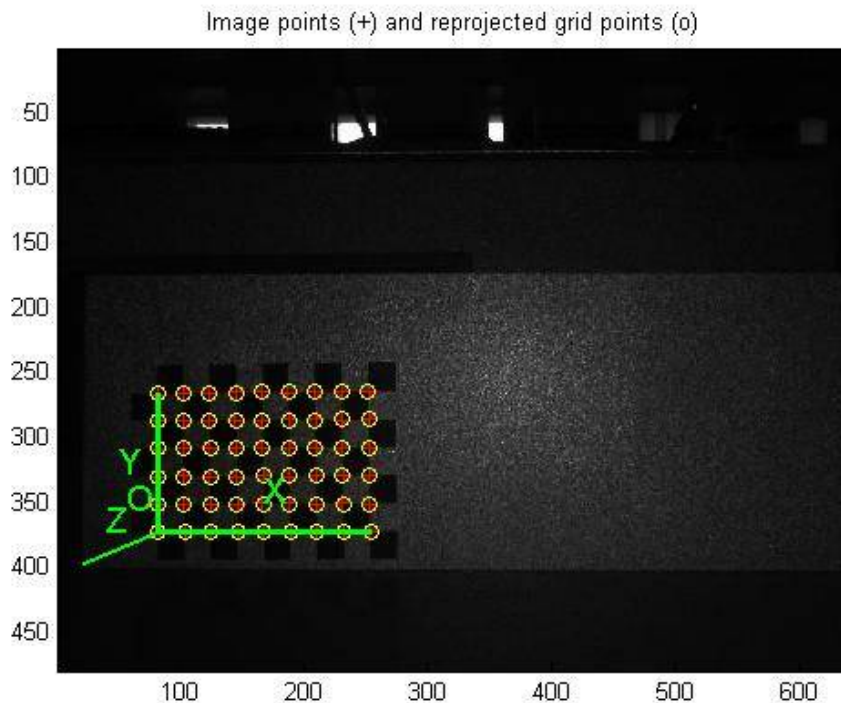


Figura 3.6: Calibração extrínseca da câmara IR

$$T_{IR} = \begin{bmatrix} -424.07 \\ 229.31 \\ 957.07 \end{bmatrix}$$

$$R_{IR} = \begin{bmatrix} 0.999854 & -0.016329 & -0.005073 \\ -0.016580 & -0.998388 & -0.054279 \\ -0.004179 & 0.054355 & -0.998513 \end{bmatrix}$$

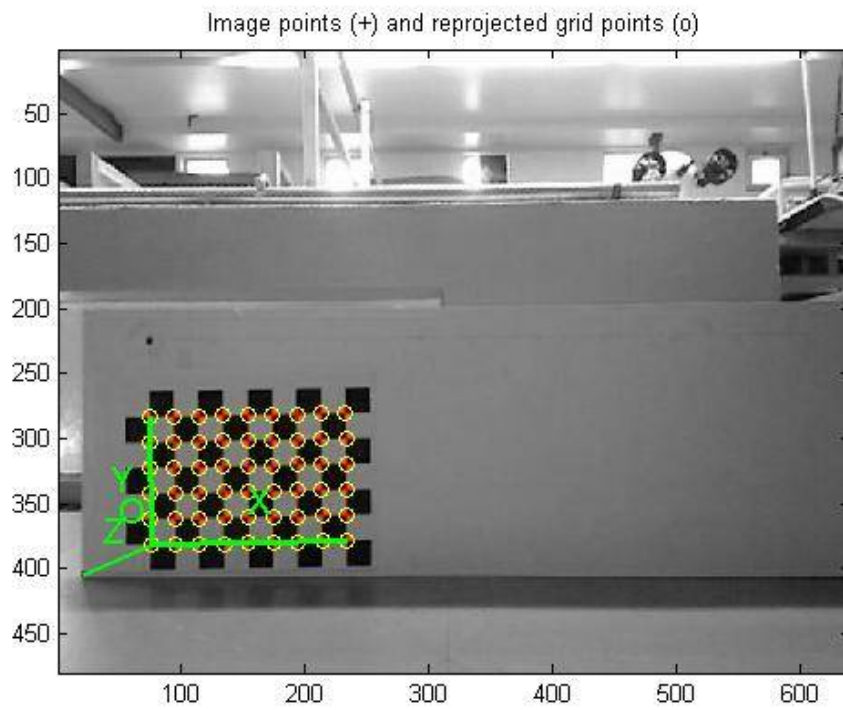


Figura 3.7: Calibração extrínseca da câmara *RGB*

$$T_{RGB} = \begin{bmatrix} -441.54 \\ 229.39 \\ 962.71 \end{bmatrix}$$

$$R_{RGB} = \begin{bmatrix} 0.999760 & -0.019715 & -0.009546 \\ -0.020051 & -0.999135 & -0.036440 \\ -0.008819 & 0.036622 & -0.999290 \end{bmatrix}$$

3.3 Medidas no mundo

Tendo todos os valores de calibração da câmara, é possível determinar as coordenadas no mundo real. No entanto, falta determinar a medida de profundidade. Esta medida é obtida após conversão dos dados do *Kinect* e retorna o valor em metros. A equação 3.11 permite obter o valor de profundidade e foi proposta por Stéphane Magnenat [31].

$$p_m = 0.1236 * \tan\left(\frac{p_k}{2841.5} + 1.1863\right) \quad (3.11)$$

Em p_m é o valor da profundidade em metros e p_k é o valor retornado pelo *Kinect*.

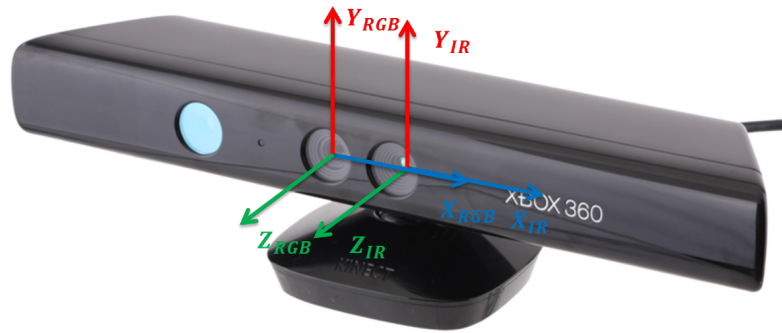


Figura 3.8: *Kinect* com referências em ambas as câmaras [8]

O primeiro passo para determinar as coordenadas no mundo é calcular as coordenadas no plano de imagem. Estas dependem dos parâmetros intrínsecos da câmara e do valor de profundidade.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} \frac{(u-c_x)*p_m(u,v)}{f_x} \\ \frac{(v-c_y)*p_m(u,v)}{f_y} \\ p_m(u,v) \end{bmatrix} \quad (3.12)$$

A equação 3.12 mostra como determinar as coordenadas no plano da imagem. Observando a mesma pode concluir-se que a origem do referencial está centrada no plano da imagem e o valor de p_m de profundidade têm a mesma direção e sentido do eixo do Z_{IR} representado na Figura 3.8.

De seguida é necessário recorrer à matriz de rotação e vetor de translação, de modo a obter as coordenadas no espaço 3D de acordo com o referencial no mundo, escolhido na calibração extrínseca. A equação 3.13 permite obter as coordenadas no mundo.

$$C_w = (C_c - T) * R \Leftrightarrow \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}^T = \left(\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}^T - \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}^T \right) * \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} \quad (3.13)$$

Observando a equação 3.13 C_c são as coordenadas da frame da câmara em $2D$, já calculadas na equação 3.12, T é o vetor de translação e R é a matriz de rotação. Todos os resultados obtidos através deste algoritmo serão mostrados no capítulo referente aos resultado.

Capítulo 4

Prototipagem da Garra

Este capítulo documenta o desenvolvimento de uma garra. A mesma foi utilizada para ensinar a um manipulador os movimentos que este deve reproduzir, bem como, enviar dados relevantes que determinam o seu bom funcionamento. Esta garra foi projetada com recurso ao software de desenho de peças 3D, *SolidWorks*. Posteriormente as peças dimensionadas foram impressas, numa impressora 3D e de seguida foi feita a sua assemblagem. Um fator importante neste capítulo é o hardware que constitui a garra. Este será responsável pela aquisição de dados relevantes para medir a força, a orientação e a abertura da garra, sendo estes fatores de realce no âmbito desta dissertação.

4.1 Estrutura da garra

A garra consiste numa adaptação de um *Mini servo gripper Plate* de dois dedos, disponível no thingiverse [32], o qual disponibiliza as peças em formato de impressão 3D. Tal como o nome indica, esse *gripper* foi criado com propósito de funcionar através de um servomotor, sendo este responsável pela abertura e fecho dos dedos. A Figura 4.1 mostra isso mesmo.

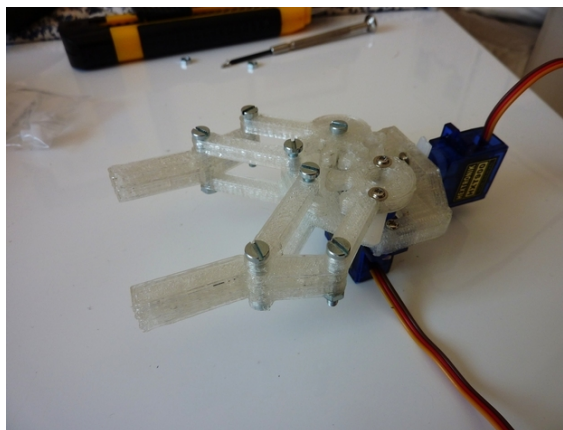


Figura 4.1: Mini servo gripper Plate

No caso aplicado, os motores desaparecem, pois a abertura e fecho de garra são feitos por um utilizador humano, responsável por esses movimentos da garra. Isto prende-se com o facto de a garra servir de auxílio à aquisição de movimento.

4.1.1 Modificações realizadas no *Mini servo gripper Plate*

A primeira alteração realizada prende-se com o ângulo de abertura da junta. Removeu-se o servomotor do desenho inicial e por sua vez, foi colocado um potenciómetro. Assim, retirou-se a abertura quadrada do servomotor da Figura 4.2 e colocou-se uma abertura arredondada em camadas para o potenciómetro fixar da melhor forma, tal como a Figura 4.3 demonstra.

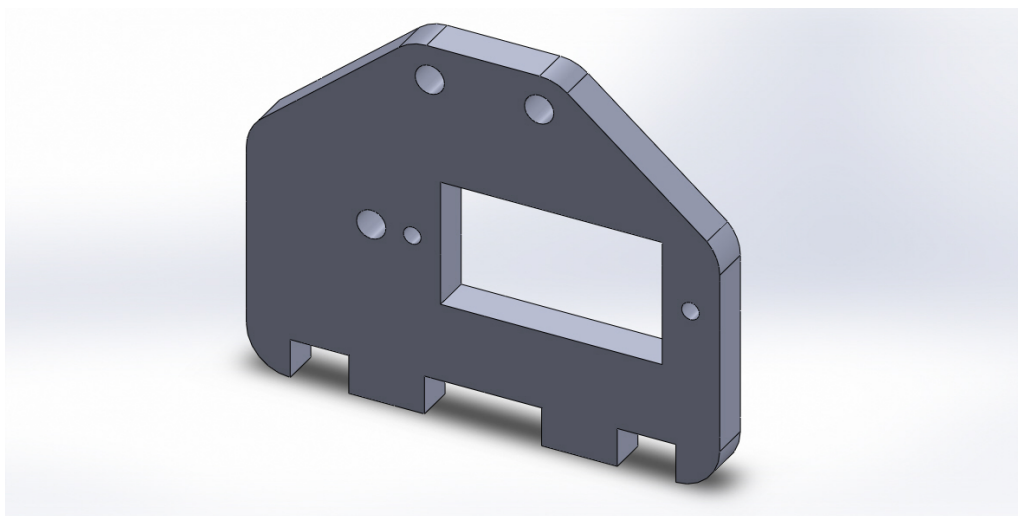


Figura 4.2: Encaixe original

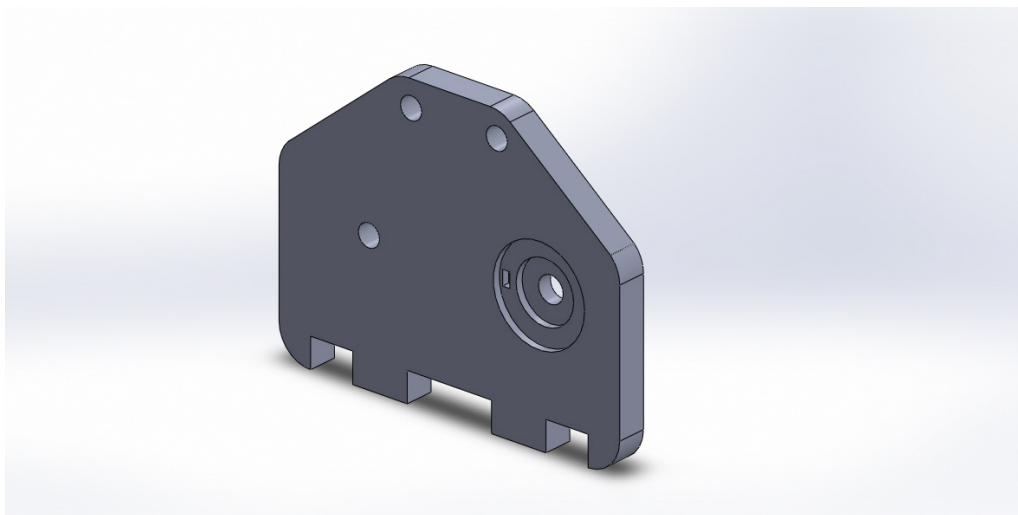


Figura 4.3: Encaixe para potenciómetro

A Figura 4.4 mostra a peça após a impressão. Esta peça já tem os furos para o suporte da caixa de hardware e para fixar o potenciômetro no seu encaixe.

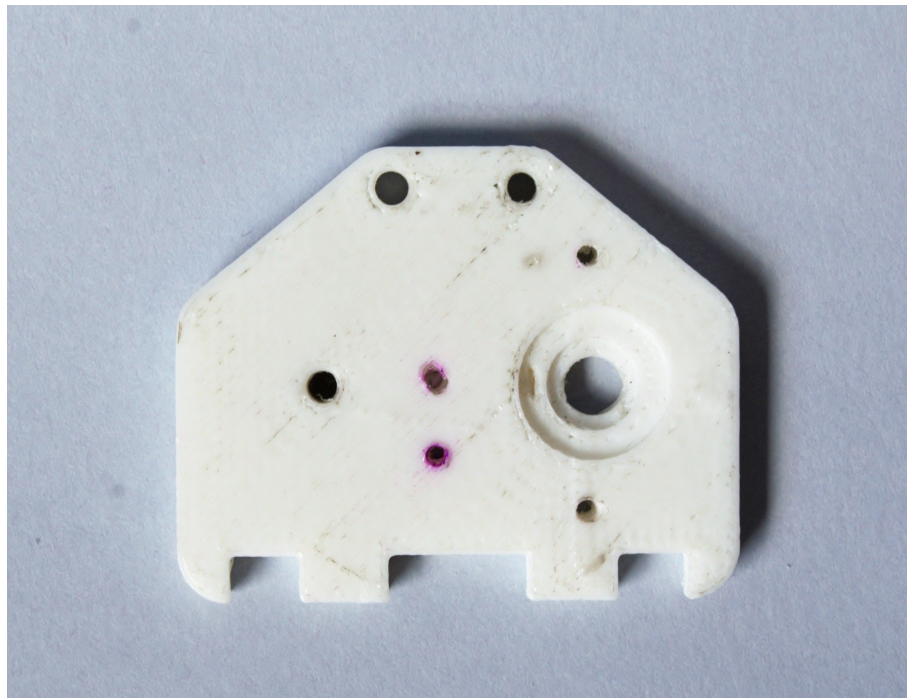


Figura 4.4: Prato da garra impresso

A engrenagem da direita do *gripper* foi alterada, já que, originalmente, o furo representado nesta serve de encaixe para a engrenagem do servo motor, mas no caso aplicado, serviu para a engrenagem do potenciômetro. As Figuras 4.5 e 4.6 demonstram as alterações realizadas, para que o veio do potenciômetro entre na engrenagem.

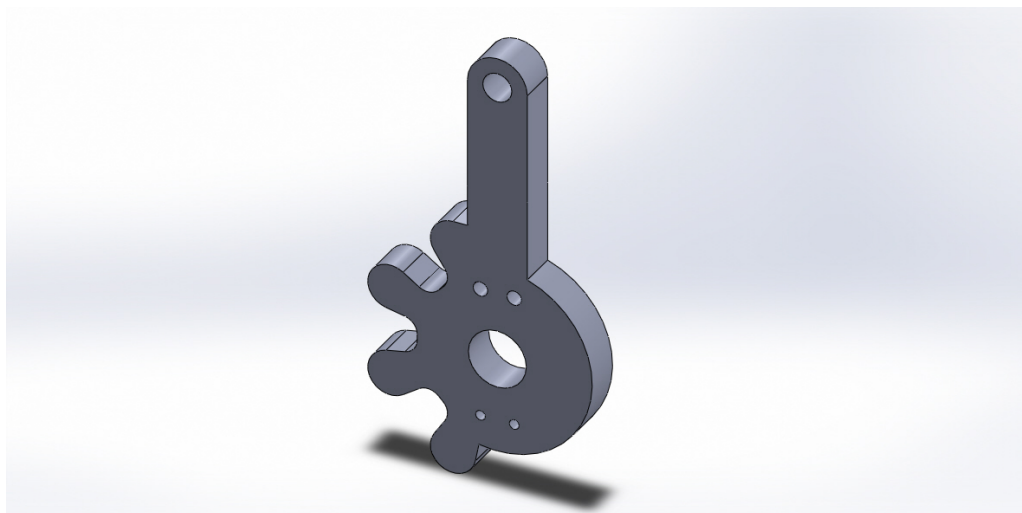


Figura 4.5: Engrenagem direita do gripper original

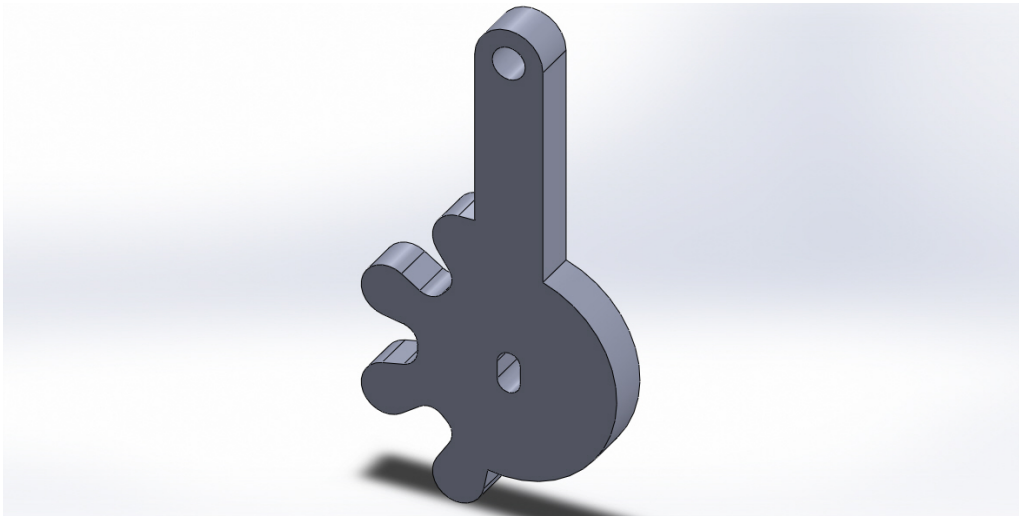


Figura 4.6: Engrenagem direita do gripper modificada

A Figura 4.7 mostra o resultado da engrenagem após impressão.



Figura 4.7: Engrenagem após impressão

O comprimento dos dedos foi aumentado, por forma a ser possível incorporar em cada dedo um sensor de força, sendo este um sensor YZC-161B [33] com gama de funcionamento de 0 a 5kg. A Figura 4.8, permite observar a diferença entre o dedo original e as alterações implementadas. Desta forma, verifica-se que o comprimento do dedo mudou da medida original de 25mm, para 55mm e que ocorreu uma translação à esquerda de 11.6mm. Possui ainda dois furos, cujos centros estão distanciados a um valor de 29mm e raio 1.25mm. Este furos foram usados para fixar a estrutura que segura os sensores de força.

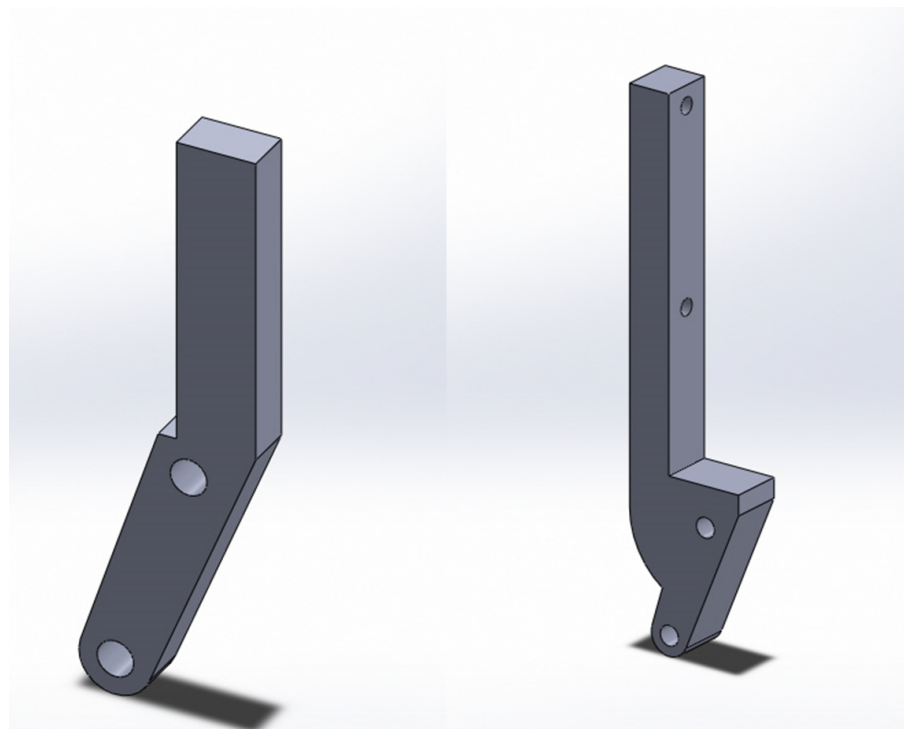


Figura 4.8: Dedo do Gripper original vs desenvolvido

4.1.2 Novas peças desenvolvidas

As novas peças desenvolvidas servem para suportar e fixar o hardware. Assim, foram impressos uma caixa para colocar o hardware, um suporte para a respetiva caixa, um encaixe para segurar o potenciômetro e um conjunto de três peças para suportar os sensores de força.

Como podemos observar na Figura 4.9 a caixa do hardware é quadrada contendo uma pequena borda interior que suporta a veroboard. Esta borda é necessária para que exista espaço para os dois parafusos que são colocados nos furos que se encontram na superfície da caixa, bem como, espaço para os pinos soldados na veroboard. Estes furos possuem um raio de 0.9 milímetros.

A caixa mede 65 milímetros de comprimento, 55 milímetros de largura e 6 milímetros de altura. A superfície interna tem como dimensões 58 milímetros de comprimento, 48 milímetros de largura e 2 milímetros de altura. A borda de suporte adiciona 2 milímetros às medidas da superfície interna, ao passo que o quadrado que representa a borda tem 60 milímetros de comprimento, 50

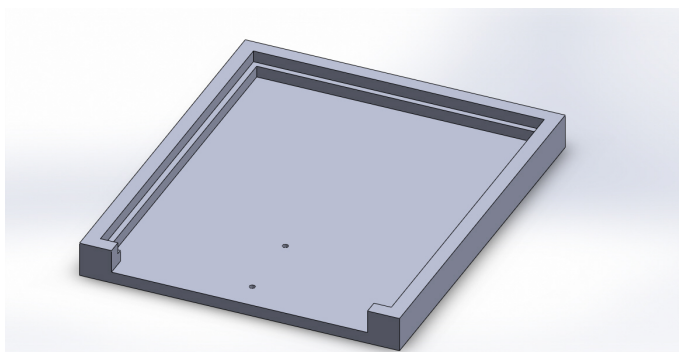


Figura 4.9: Caixa do Hardware

milímetros de largura e 4 milímetros de altura. Logo, a veroboard terá as mesmas dimensões desta borda. Na zona frontal desta caixa existe uma abertura que permite conectar os fios dos sensores de força e do potenciômetro.

O suporte criado para a caixa, consiste num pilar, com duas hastes em cada extremidade. Estas hastes possuem furos com 0.9 milímetros de raio, que servem para aparafusar uma extremidade à caixa de hardware e a outra ao prato da garra. A Figura 4.10 exibe o suporte desenvolvido. Este tem 40 milímetros de altura, 10 milímetros de espessura e 21 milímetros de comprimento.

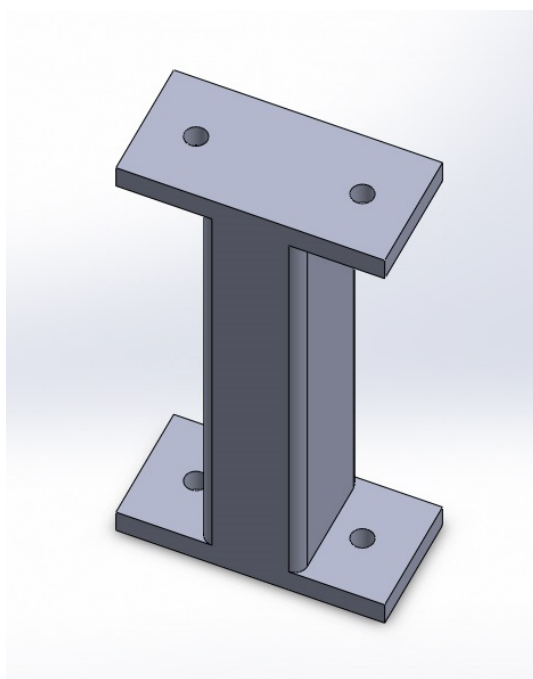


Figura 4.10: Suporte da caixa do hardware

O potenciômetro, embora tenha o seu veio na engrenagem, com o movimento de abertura e fecho da garra, pode soltar-se. Para evitar esta situação foi criado um encaixe que garante que o potenciômetro não se mova, observado na Figura 4.11.

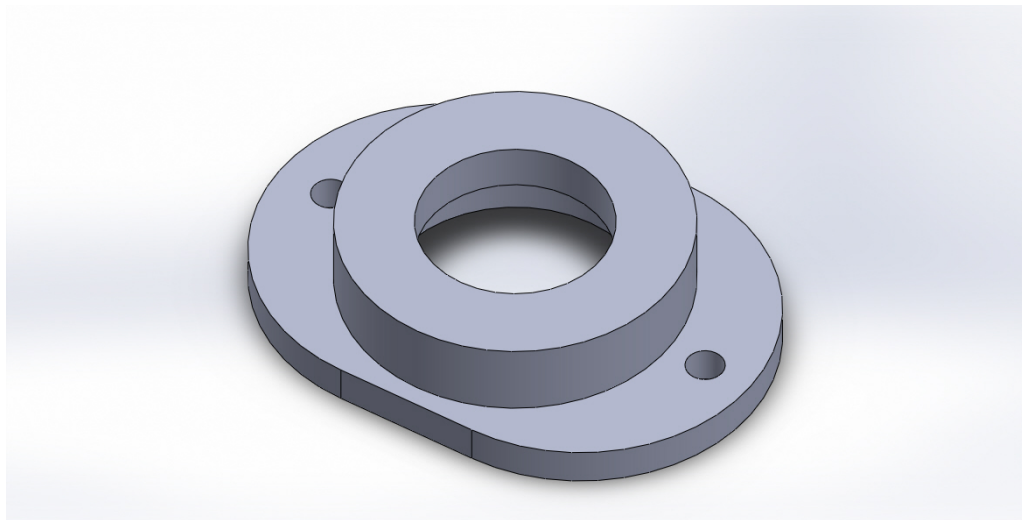


Figura 4.11: Encaixe para potenciômetro

Esta peça tem um furo central de 5 milímetros de raio, por onde os fios que vão ligar ao Arduino, podem passar. Possui ainda dois furos nas extremidades que, tal como no suporte da Figura 4.10, são usados para fixar esta peça ao prato da garra e assim prender o potenciômetro.

Para colocar o sensor no dedo são necessárias três peças: a base onde o sensor está pousado e as restantes duas peças que são aparafusadas, lateralmente, de forma a segurar o sensor. A Figura 4.12 representa a base onde o sensor estará pousado.

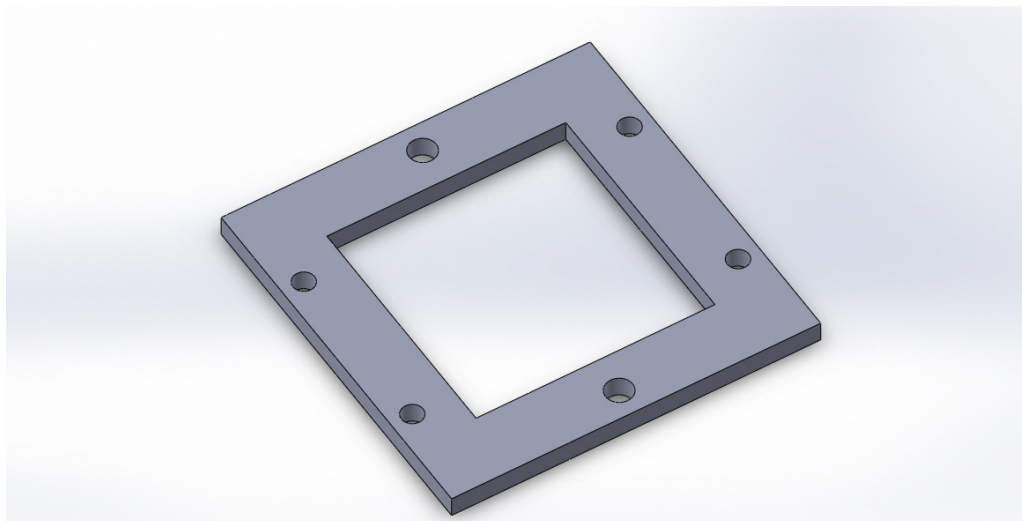


Figura 4.12: Base do sensor de força

O furo quadrado que se pode observar na Figura 4.12, tem 22 por 22 milímetros e é necessário para que ao pressionar o YZC-161B, a superfície central de metal possa mover-se. A Figura 4.13 foi a peça desenvolvida para fixar o sensor.

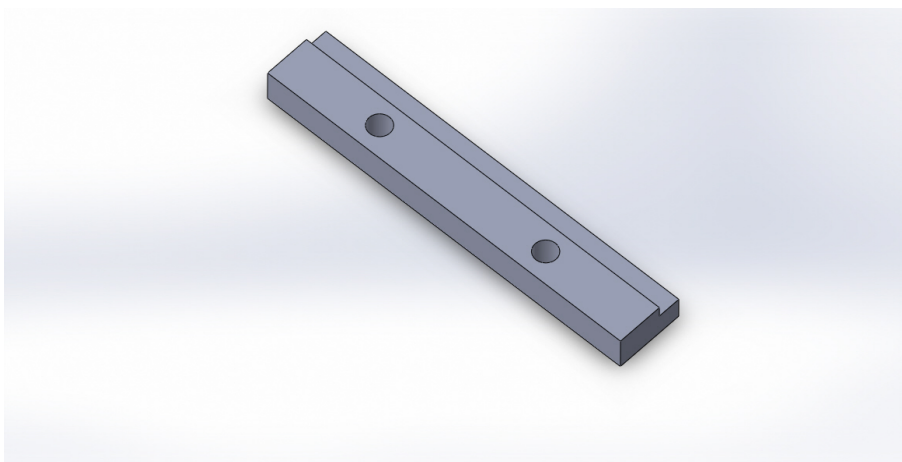


Figura 4.13: Barra de fixação do sensor de força

A Figura 4.14 mostra o resultado de como as peças das Figuras 4.12 e 4.13 após a impressão, são aglomeradas e os seus sensores acoplados.

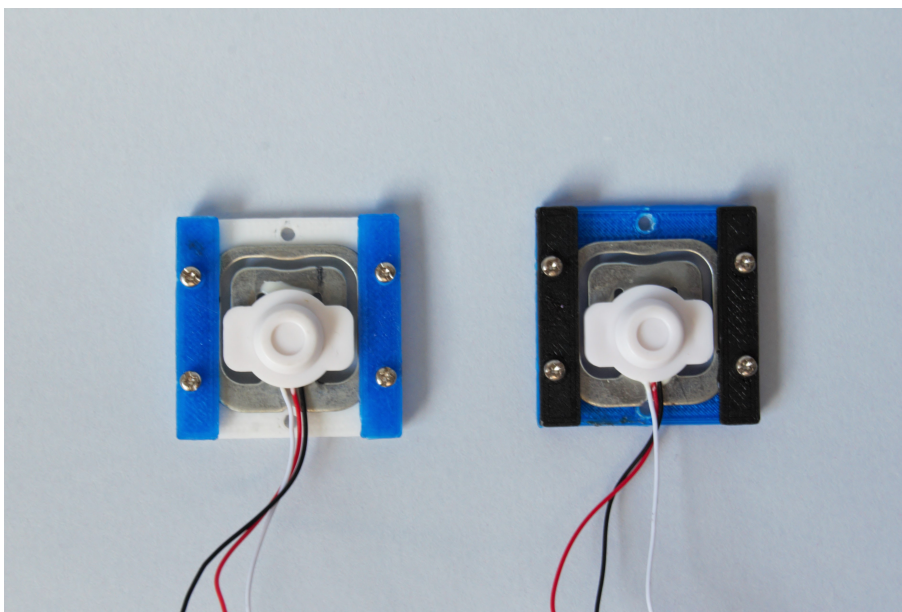


Figura 4.14: Peças impressas com sensores de força acoplados

4.1.3 Vantagens e desvantagens da impressão 3D

Todas estas peças que compõem a garra são impressas numa impressora 3D, evidenciando a vantagem do custo reduzido do material de impressão e a facilidade de correção das peças. O facto de estas serem pequenas, permite que o tempo de impressão seja reduzido, já que cada peça, individualmente, demora em média, vinte minutos a ser impressa.

A principal desvantagem da impressão 3D, é a qualidade das peças: por norma são menos robustas, comparativamente, com as peças criadas com método de injeção de molde. O tempo de impressão é também um problema, visto que peças de maiores dimensões, comparativamente, com as descritas anteriormente, demoram muito tempo a ser impressas. É necessário ter em conta o tamanho da peça a ser impressa, já que esta não pode ser maior do que a superfície de impressão. Uma outra desvantagem é o preço do equipamento de impressão, cujo valor pode rondar os 400 euros, para impressoras mais baratas e os 20000 euros, para impressoras mais caras.

Com todas as peças impressas, é possível fazer a montagem, com resultado visível na Figura 4.15.

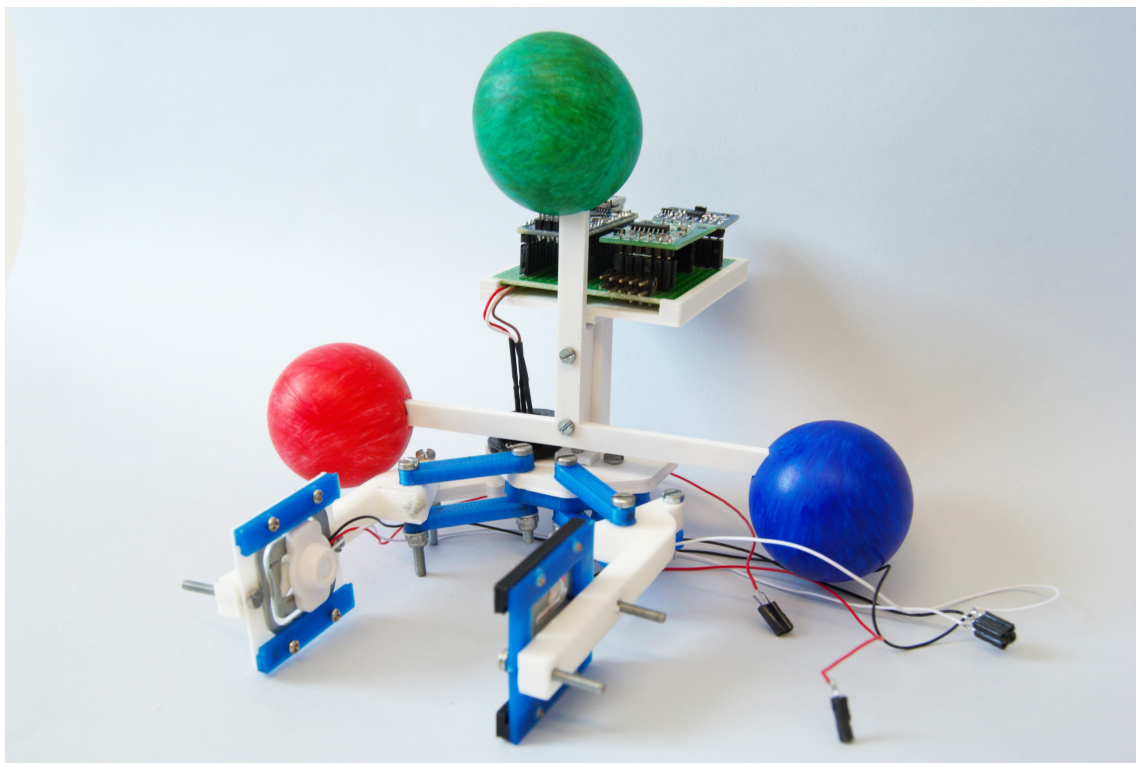


Figura 4.15: Garra completa

4.2 Sensores de força

Os sensores de força YZC-161B são sensores resistivos, cuja medida se baseia no princípio de uma ponte de wheatstone. Cada um destes sensores corresponde a meia ponte, com três fios: vermelho, branco e preto, tal como na Figura 4.16. Tipicamente, o fio vermelho corresponde ao sinal a ser enviado para o conversor. No entanto, para confirmar, mediu-se com um multímetro nos pares de terminais entre fio branco e fio vermelho, fio preto e fio vermelho e fio branco e fio preto. Entre cada um destes pares, o valor de resistência obtido foi, respetivamente, $1k\Omega$, $1k\Omega$

e $2k\Omega$. Com base nestas medidas conclui-se que o fio vermelho é de facto, o fio com o sinal de leitura de medida da carga [26].

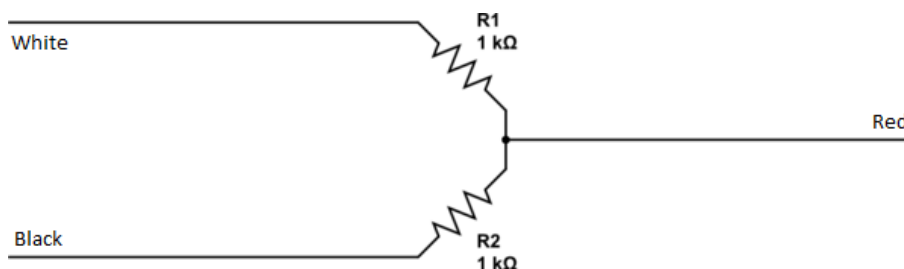


Figura 4.16: Esquema elétrico do sensor de carga [9]

4.2.1 Circuito equivalente

Como já referido, cada dedo da garra terá um sensor. Assim, o esquema de medida de carga corresponde a uma ponte de wheatstone completa, tal como demonstra a Figura 4.17. Sendo uma ponte de wheatstone, quando o sensor é pressionado a resistência varia, tal como representado na seguinte equação [34].

$$V_{out} = \left(\frac{R3}{R3 + R4} - \frac{R2}{R1 + R2} \right) * V_{in} \quad (4.1)$$

Na Figura 4.17 é possível observar os terminais E+, E-, A+ e A- que irão ligar ao conversor analógico para digital HX711.

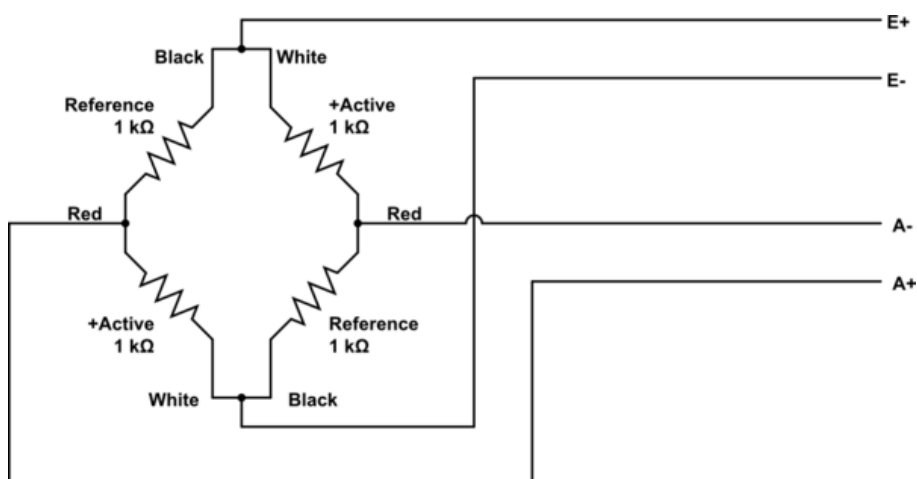


Figura 4.17: Esquema elétrico de dois sensores de carga ligados [9]

A Figura 4.18 exibe o integrado HX711 utilizado para amplificar e converter o sinal para ser lido no Arduino. Os pinos do lado esquerdo do integrado ligam aos sensores, ao passo que os do lado direito estão conectados ao Arduino.

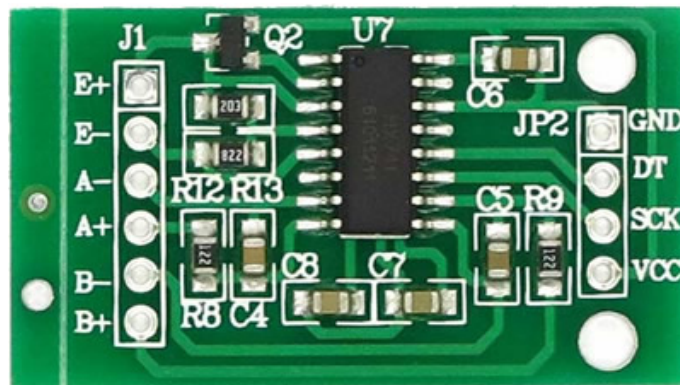


Figura 4.18: Integrado HX711 [10]

4.2.2 Vantagens do sensor

Este sensor de força tem a vantagem de ser de pequena dimensão - 26mm por 26mm e 10mm de altura - sendo estas medidas apropriadas para estarem na ponta dos dedos da garra. O custo de um sensor ronda o valor de 2,95 euros.

4.2.3 Fixação

Os sensores necessitam de ser colocados nos dedos da garra. Foi criado em *SolidWorks* um encaixe que permite fixar os sensores, tal como é possível observar na Figura 4.19.

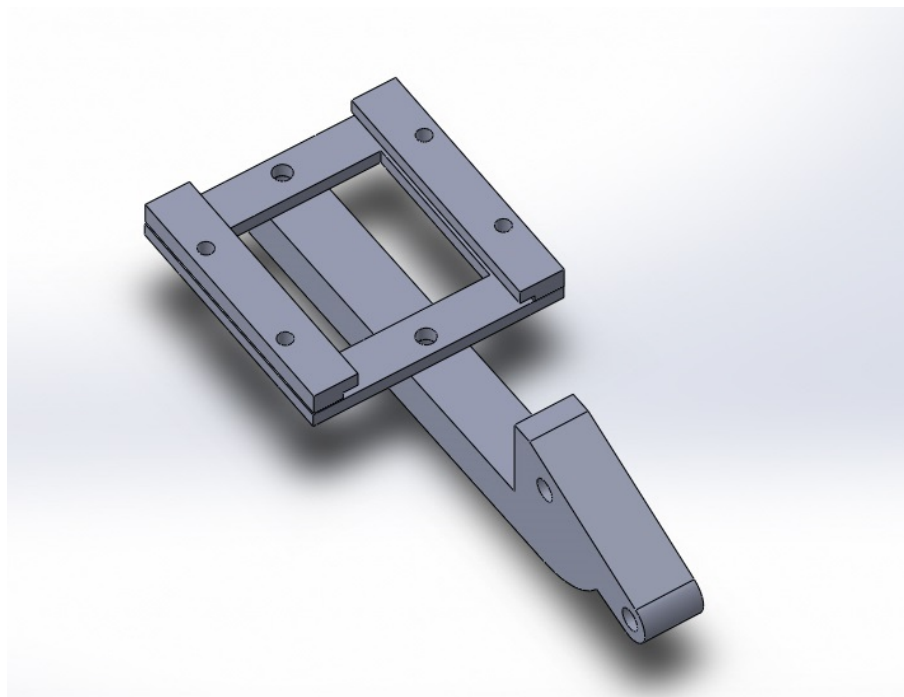


Figura 4.19: Estrutura de fixação dos sensores na garra

A Figura 4.20 mostra o resultado obtido após as peças serem impressas e fixado o sensor.

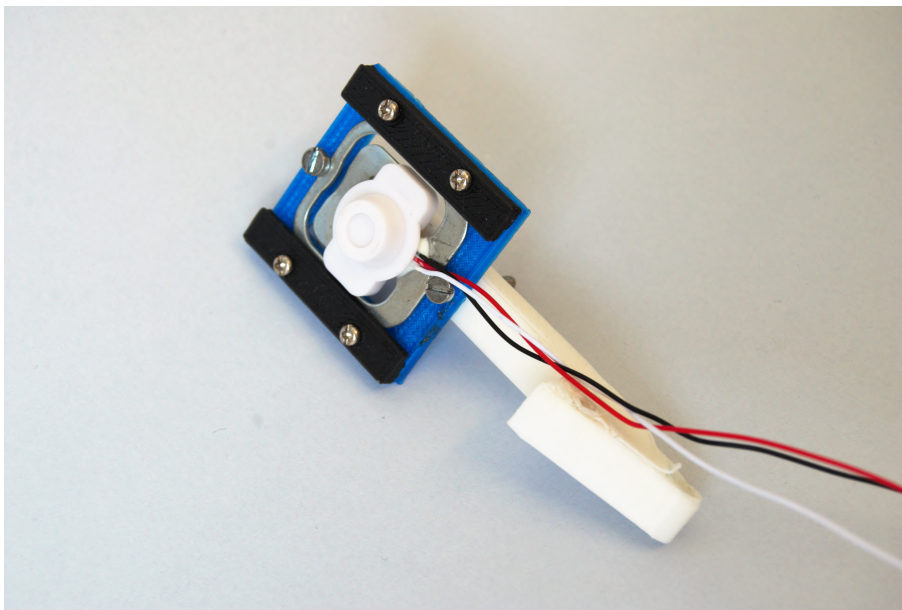


Figura 4.20: Estrutura com sensores fixados no dedo da garra

4.3 Medição de ângulo

Como já referido, para medir o ângulo de abertura da garra, colocou-se um potenciômetro no local que se destinava a um servomotor. Quando a garra abre, a engrenagem da direita move-se no sentido dos ponteiros do relógio e o potenciômetro tendo o seu veio no furo da engrenagem, sofre a mesma rotação.

4.3.1 Medidas do potenciômetro

Este potenciômetro tem valor de $5k\Omega$ e é alimentado com uma tensão de 5V. Para verificar o seu comportamento de cada vez que havia variação na abertura e fecho da garra, elaborou-se um conjunto de medidas que serviram para obter o seguinte gráfico.

Observando o gráfico da figura 4.21, verifica-se que o comportamento deste potenciômetro é linear com variação da resistência a ter poucas oscilações. Desta forma, as medidas obtidas a partir do potenciômetro, são bastante precisas.

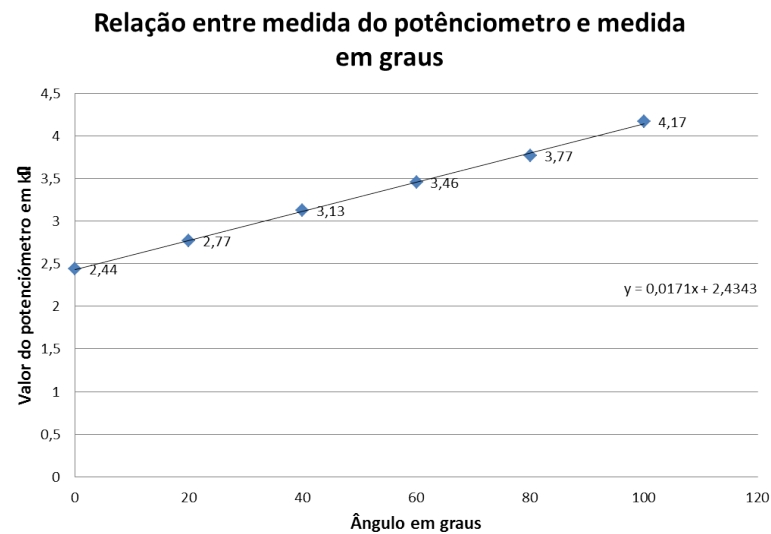


Figura 4.21: Relação entre medida do potenciômetro e medida em graus

A utilização do potenciômetro revelou-se bastante acertada e precisa, bem como, barata e fácil de adquirir.

4.3.2 Fixação

O potenciômetro é fixado nas peças representadas nas Figuras 4.3 e 4.11. O resultado obtido após a impressão 3D e montagem das peças pode ser visto na Figura 4.22.

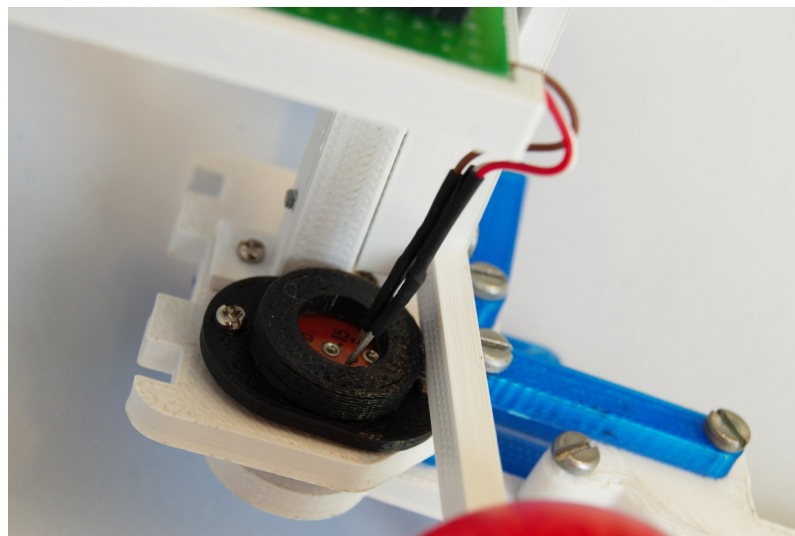


Figura 4.22: Potenciômetro fixado na garra

4.4 Acelerómetro

O integrado utilizado, o MPU6050, é constituído por um acelerómetro e um giroscópio de três eixos cada, conferindo assim seis graus de liberdade. Este tipo de equipamento permite determinar a orientação da garra e desta forma estimar a posição da mesma.

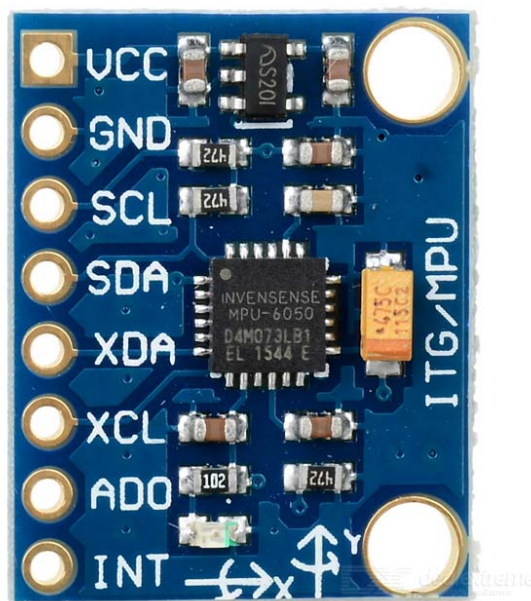


Figura 4.23: Integrado do MPU6050 [11]

Na Figura 4.23 podemos observar o integrado do MPU6050. Os pinos VCC, GND, SCL (Serial Data) e SDA (Serial Clock) ligam ao Arduino.

4.4.1 Princípio de funcionamento

O acelerómetro interno do MPU6050 utiliza 3 pontos de massa (um em cada eixo), fazendo variar estes pontos, de forma a estimar a aceleração. Para verificar se as medidas estão corretas, é necessário colocar o acelerómetro numa superfície plana e verificar se os valores obtidos correspondem a 0g para os eixos X e Y e 1g no eixo do Z. A escala de valores da saída digital pode ser seleccionada na seguinte gama de valores: ± 2 , ± 4 , ± 8 e $\pm 16g$ [12].

Como já referido, este integrado possui um giroscópio, que permite detetar a a rotação em cada um dos eixos X, Y e Z. Quando sofre uma rotação, o giroscópio, independentemente do eixo, o efeito Coriolis causa uma vibração que é detetada. Este sinal detetado produz uma tensão proporcional ao ângulo. Por último, esta tensão é convertida por um ADC de 16 bits, programável para converter desde 8000 amostras até 3.9 amostras por segundo. A escala de valores pode ser escolhida dentro da seguinte gama: ± 250 , ± 500 , ± 1000 e ± 2000 graus por segundo [12].

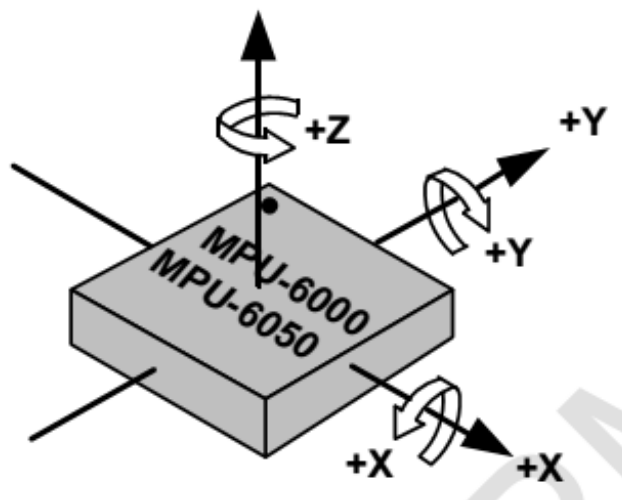


Figura 4.24: Eixos do MPU6050

Na Figura 4.24 podemos observar onde estão definidos os eixos X, Y e Z, do MPU6050. O ponto no canto superior esquerdo representa o pino 1 do integrado.

4.4.2 Comunicação

A comunicação deste equipamento é feita segundo o protocolo de comunicação I2C. Assim o Arduino recebe os valores a analisar através dos pinos *SDA* e *SCL*. A comunicação estabelecida é do tipo *Master - Slave*, onde o *Master* coloca o endereço do *Slave* na trama. Caso o endereço corresponda ao do *Slave*, este reconhece o *Master*. O endereço do *Slave* do MPU6050 é b110100X de 7 bits [12].

A comunicação I2C funciona de acordo com condições de *Start* e *Stop*. Quando o *Master* ativa a condição *Start* ocorre uma transição na porta *SDA*. Esta muda de *HIGH* para *LOW*, enquanto a *SCL* mantém o seu valor a *HIGH*. Quando a condição *Stop* é ativa a porta *SDA* muda de *LOW* para *HIGH* e *SDA* mantém o seu valor *HIGH*. A Figura 4.25 retirada da *datasheet* do MPU6050 apresenta o procedimento de comunicação.

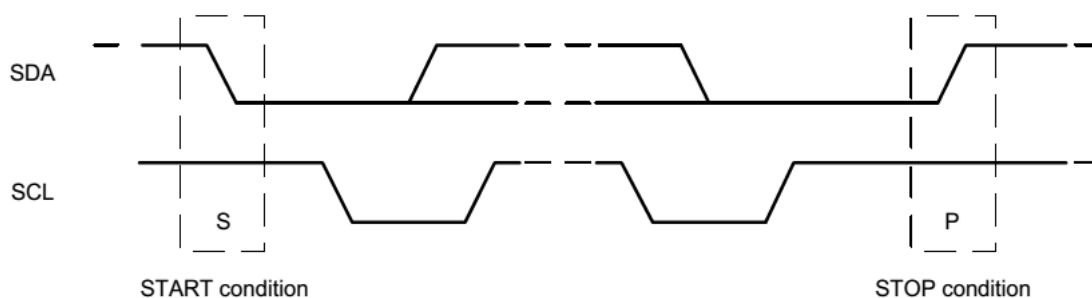


Figura 4.25: Start e Stop da comunicação I2C [12]

Após a condição *Start* ser ativada o *Master* envia o endereço de 7 bits ao *Slave* e um oitavo bit de leitura/escrita. Este oitavo bit indica se o *Master* escreve ou recebe informação do *Slave*. De seguida o *Master* liberta a trama *SDA* e aguarda pelo sinal *ACK*.

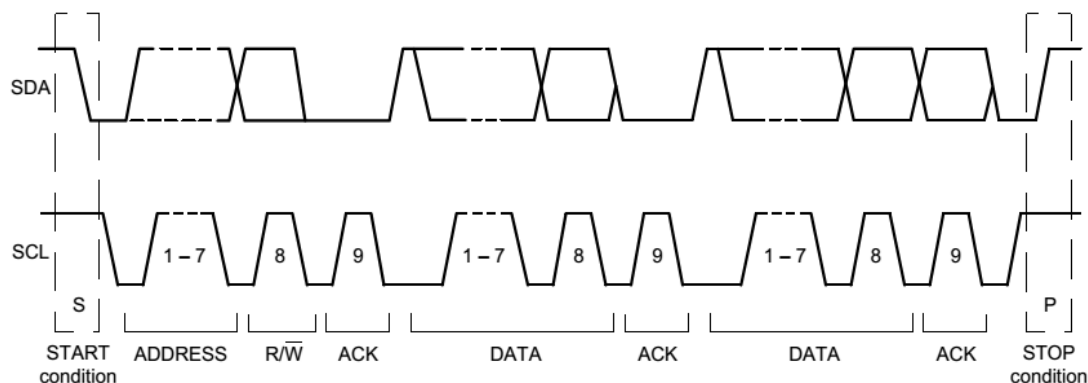


Figura 4.26: Transmissão de informação do protocolo I2C [12]

Como podemos observar na Figura 4.26 a cada 8 bits transmitidos segue o bit correspondente ao ACK: aqui a porta *SDA* muda para *LOW* e a porta *SCL* toma valor *HIGH*. O valor de *SDA* muda quando *SCL* recebe novamente o endereço. A transferência de informação termina quando *SDA* e *SCL* tem valor *HIGH*.

4.4.3 Vantagens da utilização do MPU6050

Este equipamento é apropriado para o efeito desejado, pois o integrado tem os componentes necessários para medir a aceleração nos três eixos e é um equipamento barato, com preço aproximado de 5.35 euros. O MPU6050 encontra-se fixo na veroboard, junto do restante hardware.

4.5 Marcadores de cor

A utilização de marcadores de cor ajuda a detetar a garra, visto que o *Kinect* não consegue detetar da melhor forma as coordenadas no mundo real da garra. Estes marcadores têm a particularidade de serem grandes e de cor vermelho, verde e azul. A utilização destas três cores facilitam a sua identificação, pois a câmara é a cores.

4.5.1 Fixação

Para a sua fixação foi criada uma estrutura em *SolidWorks*, que mais tarde foi impressa em 3D, em forma de T. Em cada ponta desta peça encontram-se uma esfera, de 2 centímetros de raio, com a respetiva cor. A Figura 4.27 mostra o resultado final da implementação dos marcadores de cor na garra.

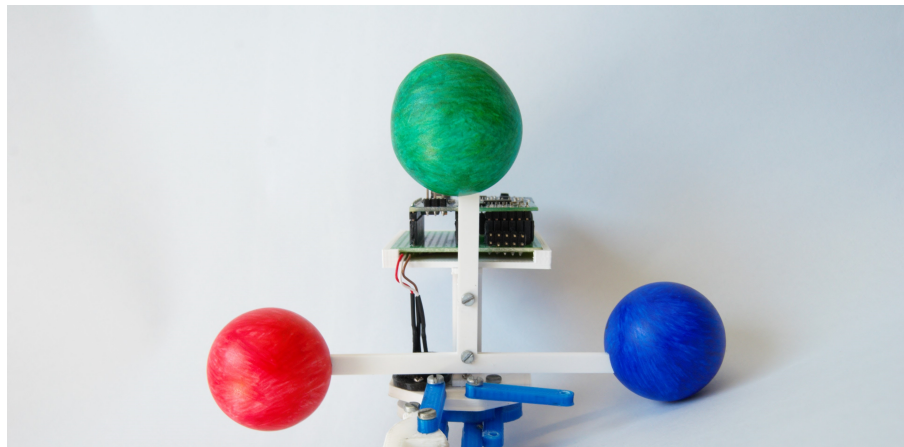


Figura 4.27: Garra com marcadores RGB

4.6 Plataforma de desenvolvimento

O Arduino nano, foi escolhido para fazer a aquisição e processamento de sinal, pois tem a vantagem de ser pequeno e leve, comparativamente com o Arduino uno. Este Arduino tem de comprimento 45 milímetros, de largura 18 milímetros e 5 gramas de peso, enquanto que o Arduino uno tem comprimento de 68.6 milímetros, largura 53.4 milímetros e peso 25 gramas. Outro motivo desta escolha prende-se com o facto de o Arduino nano possuir o mesmo microcontrolador que o Arduino uno, o ATMEGA328.

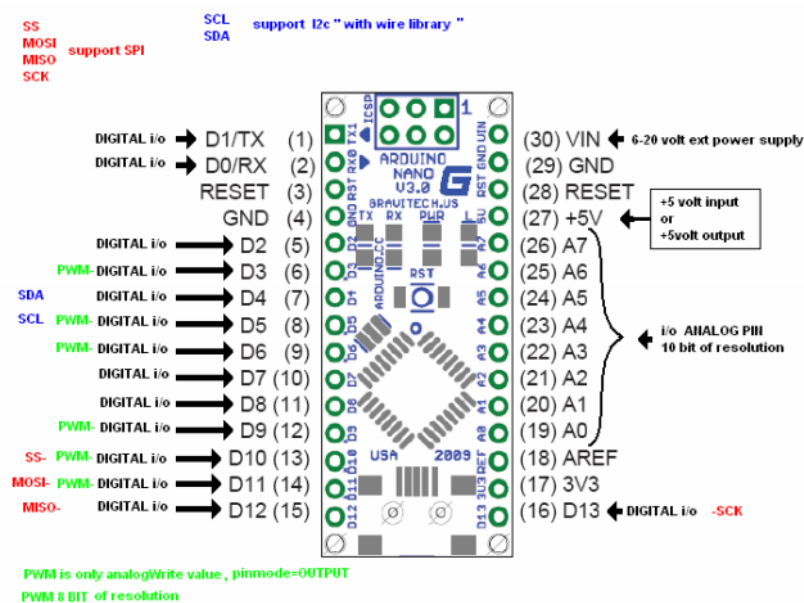


Figura 4.28: Arduino nano

Na Figura 4.28, estão representados os pinos do Arduino nano. Este contém oito pinos analógicos, que são utilizados para receber o sinal de cada um dos componentes de medida da garra. O pino A0 recebe o sinal do potenciômetro, o A2 e A3 o valor do HX711 e o A4 e A5 do MPU6050.

4.7 Interface Gráfica

Neste sistema desenvolvido é necessário criar uma interface que receba os valores dos parâmetros de calibração necessários ao cálculo das medidas do espaço 3D e os valores do todo o *hardware*, já descrito. Esta interface permite visualizar as imagens adquiridas pelo *Kinect*, tanto as *RGB* como as *IR*.

Esta interface foi desenvolvida em Lazarus, uma plataforma livre, cuja linguagem de programação é o FreePascal. Esta plataforma funciona nos diversos sistemas operativos, como Windows, Linux e MacOS. Adicionalmente tem como vantagem o facto de já existirem componentes de Lazarus a comunicar com a biblioteca *libfreenect* [35], implicada no funcionamento do *Kinect*. Estes componentes instalados fazem parte da biblioteca *5dpo* [36].

4.7.1 Interface ActiveKinect

A Figura 4.29, mostra a janela principal da interface desenvolvida. Esta ativa o *Kinect*, mostrando as imagens adquiridas do mesmo.

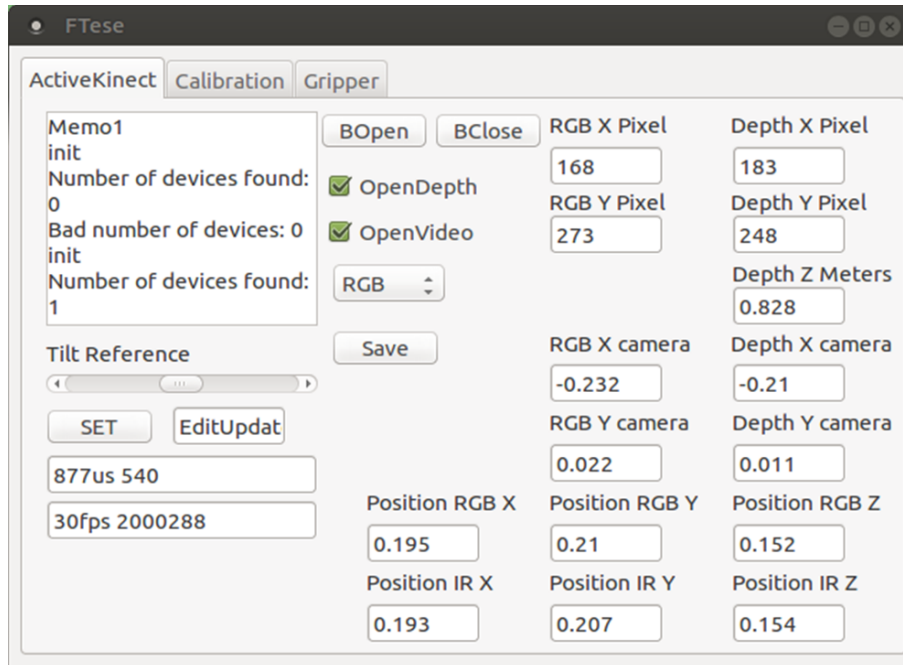


Figura 4.29: ActiveKinect

Esta janela recorre a várias funcionalidades, tais como:

- Visualizar imagem: a)RGB b)IR;

- Visualizar a imagem de profundidade;
- Guardar as imagens num formato desejado: PNG ou JPG;
- Regular a inclinação da câmara através do motor;
- Observar as coordenadas da *frame* em pixel;
- Observar as coordenadas da *frame* em metros;
- Observar as coordenadas do espaço *3D* em metros;
- Varia a taxa de atualização: botão "SET".

Todas as coordenadas apresentadas, correspondem a ambas as câmaras, *RGB* e *IR*. Quando a aplicação está ativa, o nível de inclinação da câmara transfere-se para um valor de referência, podendo a partir daí ser ajustado pelo utilizador. Nesta mesma situação, as imagens de profundidade e de vídeo estão ativas.

4.7.2 Interface Calibration

Na Figura 4.30, pode-se observar a função *Load*. Esta função permite seleccionar ficheiros XML, com nome "RGB.xml" ou "IR.xml", correspondentes aos resultados de calibração das câmaras *RGB* e *IR*, respetivamente.

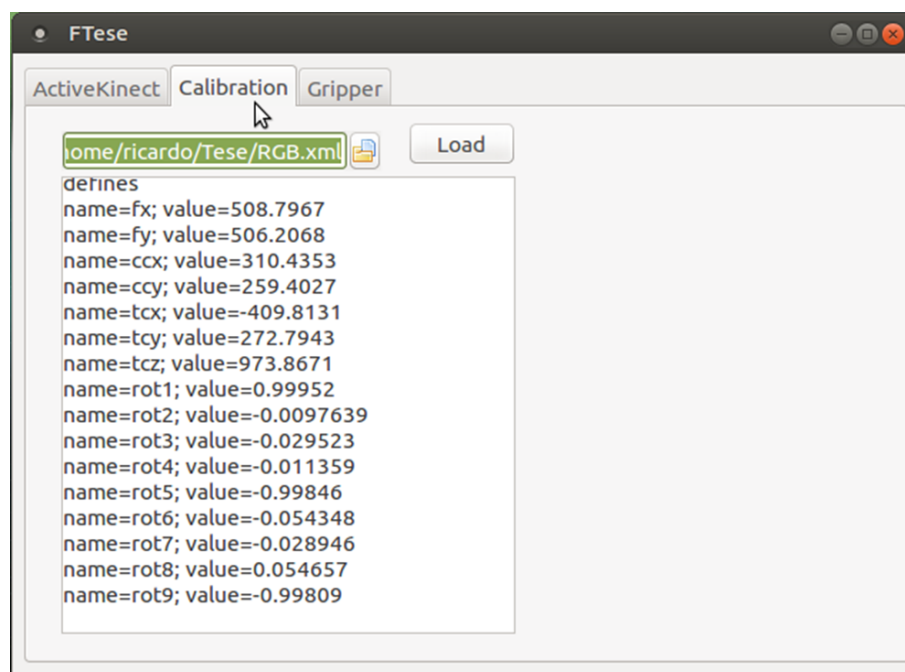


Figura 4.30: ActiveKinect

As funcionalidades da janela de calibração, são bem mais simples comparativamente com a janela *ActiveKinect*, pois esta apenas permite receber e observar os valores de calibração.

- *Load*, dos parâmetros de calibração;
- Observar os parâmetros.

4.7.3 Interface *Gripper*

Na Figura 4.31, é possível observar a janela da interface que recebe os valores de todo o hardware. Para esta funcionalidade é necessário abrir a porta série do Arduino e receber os valores do acelerômetro, sensores de força e ângulo de abertura.

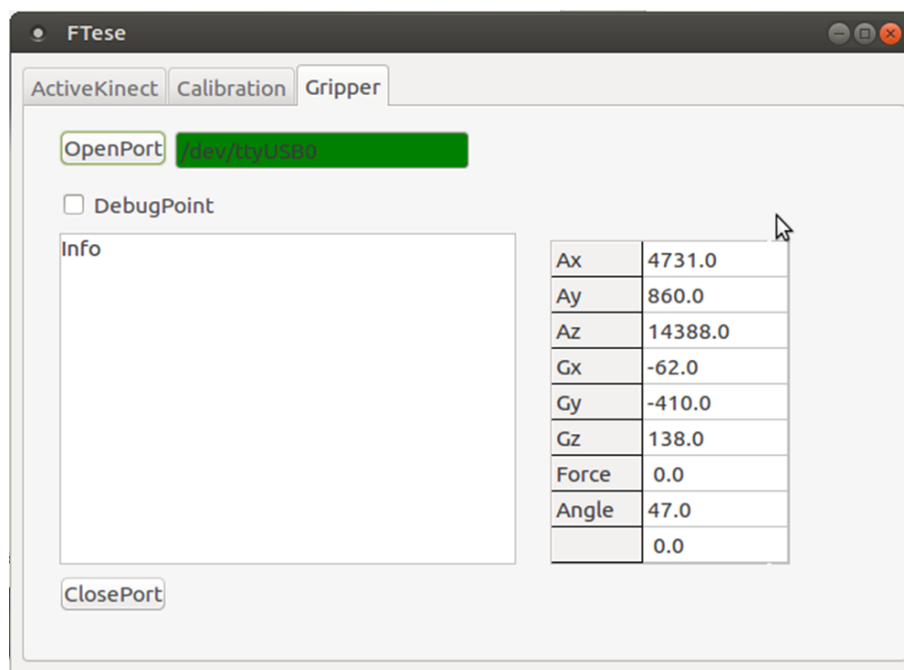


Figura 4.31: ActiveKinect

As funcionalidade desta janela são as seguintes:

- Botão *OpenPort*, para abrir a porta série com o Arduino;
- Tabela, à direita com os valores do *hardware*;
- Botão *DebugPoint*, funcionamento ingual ao Serial Monitor do Arduino IDE.

4.8 Conclusões

Neste capítulo foram descritos todos os módulos de prototipagem desenvolvidos. Adicionalmente, foi evidenciando o propósito do desenvolvimento de cada um desses módulos. Compreendeu-se que a criação da garra traz vantagens como auxiliar o utilizador a mostrar os movimentos a ser realizados. Por outro lado, o hardware de medida adicionado nesta é importante para uma maior precisão das medidas de posição da garra.

Seguidamente, descreveu-se a interface gráfica que tem a função de receber e mostrar ao utilizador, todas as informações medidas nos diferentes componentes. Esta interface é, assim, o sistema de monitorização do sistema já preparado para o seu funcionamento. Os resultados da calibração e dos testes obtidos serão descritos no capítulo seguinte.

Resultados

5.1 Medidas de profundidade do Kinect

Medidas de Profundidade Kinect vs Reais

Medidas de profundidade em metros (Real)	Resultado medido da profundidade (Real)	Resultado medido da profundidade (Kinect)
0,5	0,4960937	0,5
0,6	0,6054687	0,6
0,7	0,7070312	0,7
0,8	0,8085937	0,8
0,9	0,9101562	0,9
1,0	1,0117187	1,0
1,1	1,1132812	1,1
1,2	1,2148437	1,2
1,3	1,3164062	1,3
1,4	1,4179687	1,4
1,5	1,5195312	1,55

43

Ao observar o gráfico da Figura 5.6, conclui-se que à medida que a distância aumenta o valor medido por parte do *Kinect* aumenta também.

5.2 Coordenadas no mundo medidas

Na Tabela 5.1 e 5.2 encontram-se os resultados nas medidas no mundo real, da câmara de *IR* e da câmara *RGB*, respetivamente, determinadas com ajuda do *Kinect*. As unidades de medida desta tabela encontram-se em centímetros. O referencial corresponde ao definido na imagem do capítulo 3, da calibração extrínseca. As medidas obtidas representam 4 objetos colocados nas coordenadas indicadas nas linhas da tabela com nome "Real". Este objetos, nos restantes testes efetuados, não mudam de posição.

Tabela 5.1: Primeiro teste de medidas do mundo real da câmara da *IR*

Câmara Infravermelho		X	Y	Z
Objeto A	Real	20	22	20
	Medido	19.9	21.3	14.0
Objeto B	Real	45	35	30
	Medido	45.4	34.2	29.1
Objeto C	Real	45	35	20
	Medido	45.0	34.6	17.4
Objeto D	Real	70	22	20
	Medido	70.4	20.9	15.6

Tabela 5.2: Primeiro teste de medidas do mundo real da câmara da *RGB*

Câmara Colorida		X	Y	Z
Objeto A	Real	20	22	20
	Medido	19.7	21.5	13.9
Objeto B	Real	45	35	30
	Medido	45.2	34.0	27.6
Objeto C	Real	45	35	20
	Medido	45.9	34.2	16.7
Objeto D	Real	70	22	20
	Medido	70.5	21.1	15.6

Ao observar as tabelas de resultados das duas câmaras verificamos que as medidas retiradas do *Kinect* nos eixos X e Y são bastante próximas, enquanto que a medida no eixo do Z é algo elevada, podendo ultrapassar os 5 centímetros.

Para o segundo teste a ser efetuado mudou-se a posição da câmara, mantendo o padrão de xadrez fixo. Em relação à origem do referencial escolhido a câmara encontrava-se a 66 centímetros à direita, à altura de 64 centímetros e a 1 metro de distância. O referencial escolhido, na câmara *IR*, para este teste está representado na Figura 5.2.

Os resultados dos parâmetros extrínsecos de calibração resultam na no vetor de translação T_{IR} e na matriz de rotação R_{IR} que se seguem.

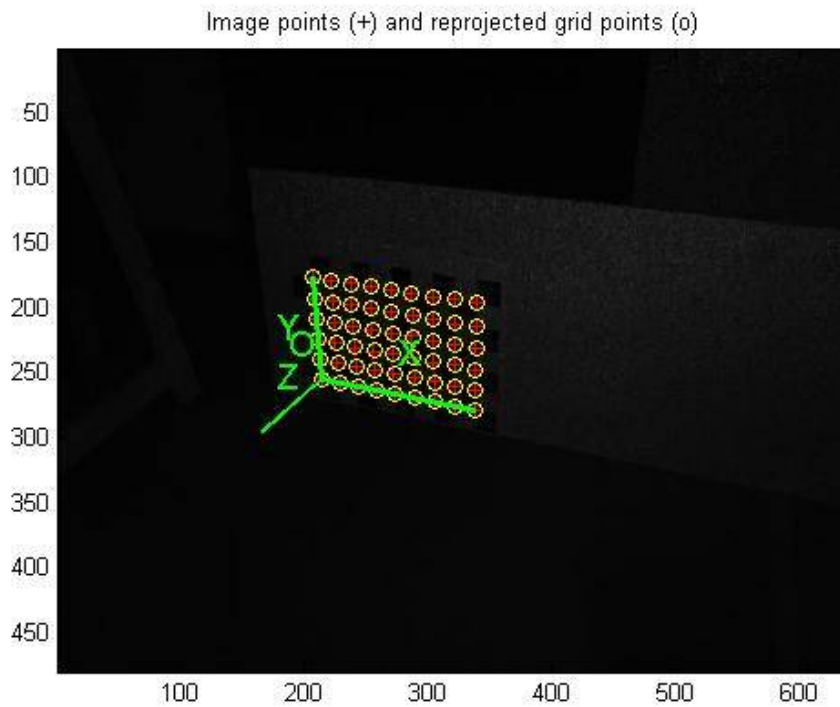


Figura 5.2: Calibração extrínseca da câmara *IR*, numa segunda posição

$$T_{IR} = \begin{bmatrix} -258.62 \\ 29.97 \\ 1275.79 \end{bmatrix}, \quad R_{IR} = \begin{bmatrix} 0.936612 & 0.005602 & -0.350324 \\ 0.164090 & -0.890452 & 0.424464 \\ -0.309569 & -0.455043 & -0.834927 \end{bmatrix}$$

Após obter os parâmetros extrínsecos mediram-se as coordenadas dos 4 objetos, já anteriormente mencionados. A Tabela 5.3 mostra os resultados da medidas no mundo da câmara *IR*.

Tabela 5.3: Segundo teste de medidas do mundo real da *IR*

Câmara Infravermelho		X	Y	Z
Objeto A	Real	20	22	20
	Medido	18.3	18.1	15.1
Objeto B	Real	45	35	30
	Medido	45.3	32.0	27.8
Objeto C	Real	45	35	20
	Medido	44.2	31.4	16.7
Objeto D	Real	70	22	20
	Medido	68.9	17.9	14.1

O referencial escolhido, na câmara *RGB*, é o mesmo do câmara *IR*. A Figura 5.3 mostra o referencial definido.

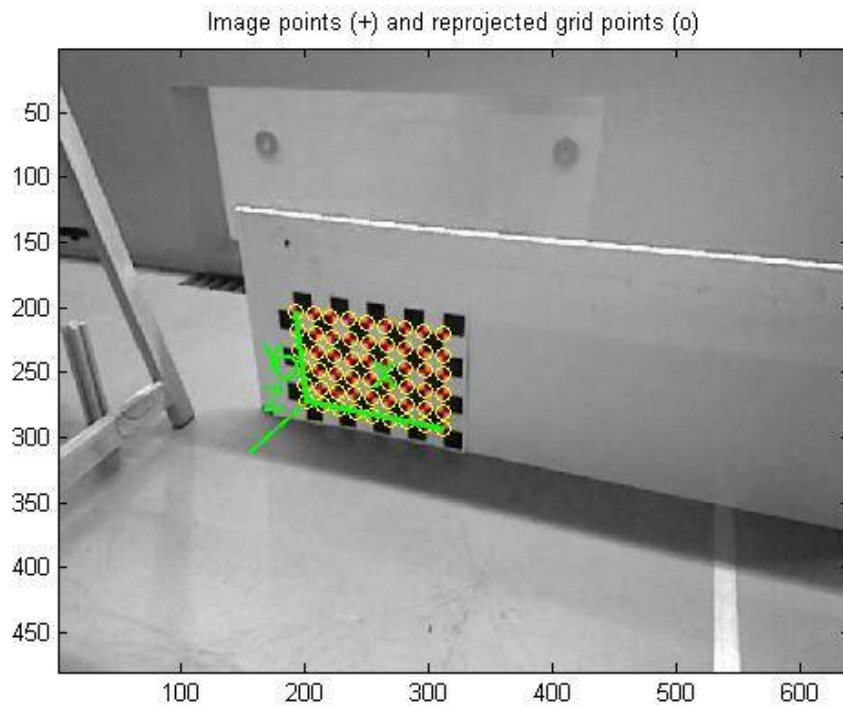


Figura 5.3: Calibração extrínseca da câmara *RGB*, numa segunda posição

Os resultados dos parâmetros extrínsecos de calibração resultam no vetor de translação T_{RGB} e na matriz de rotação R_{RGB} que se seguem.

$$T_{RGB} = \begin{bmatrix} -276.06 \\ 30.78 \\ 1277.67 \end{bmatrix}, \quad R_{RGB} = \begin{bmatrix} 0.936798 & -0.001348 & -0.349867 \\ 0.160404 & -0.887050 & 0.432913 \\ -0.310933 & -0.461672 & -0.830770 \end{bmatrix}$$

A Tabela 5.4, representa as medidas obtidas no referencial do mundo da câmara *RGB*.

Tabela 5.4: Segundo teste de medidas do mundo real da câmara da *RGB*

Câmara Colorida		X	Y	Z
Objeto A	Real	20	22	20
	Medido	18.4	18.6	14.6
Objeto B	Real	45	35	30
	Medido	45.7	32.4	27.4
Objeto C	Real	45	35	20
	Medido	44.7	31.5	16.2
Objeto D	Real	70	22	20
	Medido	69.9	18.4	13.5

Observando os resultados obtidos, conclui-se que há um maior erro nos objetos que estão mais afastados. Daí as medidas obtidas no eixo X e Y do objeto A já não serem tão próximas das reais.

Neste terceiro teste colocou-se o *Kinect* a uma menor distância do referencial do padrão de xadrez. Assim a altura e distância do *Kinect* ao padrão de xadrez mantiveram-se a 68 centímetros e 1 metros, respetivamente, mas este apenas estava transladado 25 centímetros à direita. A Figura 5.4 mostra onde foi colocado o referencial no mundo da câmara *IR*.

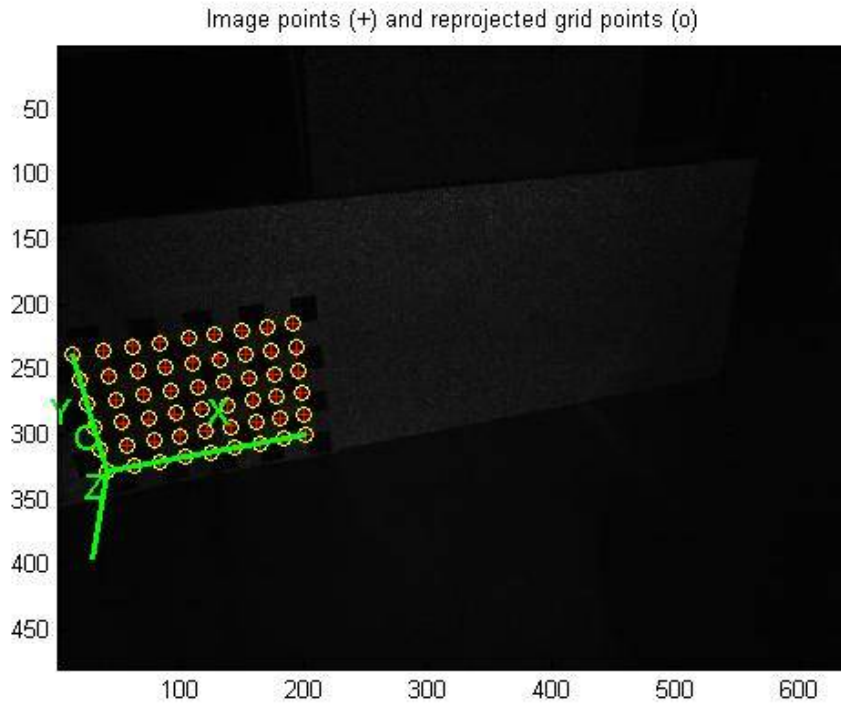


Figura 5.4: Calibração extrínseca da câmara *IR*, numa terceira posição

Os resultados dos parâmetros extrínsecos de calibração resultam na no vetor de translação T_{IR} e na matriz de rotação R_{IR} que se segue.

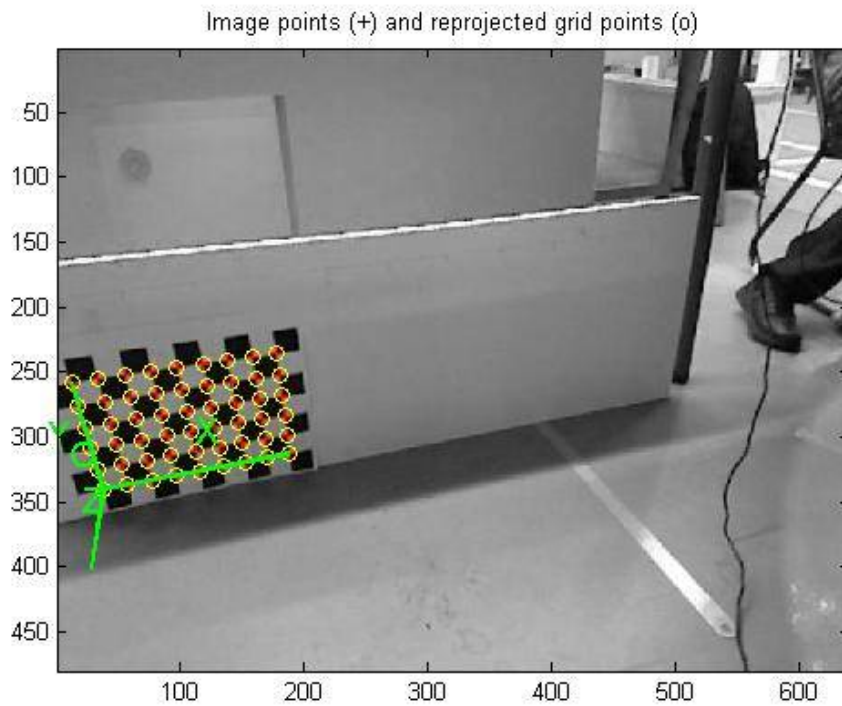
$$T_{IR} = \begin{bmatrix} -528.99 \\ 162.03 \\ 1022.77 \end{bmatrix}, \quad R_{IR} = \begin{bmatrix} 0.941384 & 0.001185 & 0.337333 \\ -0.153678 & -0.888691 & 0.431985 \\ 0.300297 & -0.458504 & -0.836418 \end{bmatrix}$$

A Tabela 5.5, representa as medidas obtidas no referencial do mundo da câmara *IR*, deste terceiro ensaio.

Tabela 5.5: Terceiro teste de medidas do mundo real da câmara da infravermelho

Câmara Infravermelho		X	Y	Z
Objeto A	Real	20	22	20
	Medido	19.3	18.1	13.2
Objeto B	Real	45	35	30
	Medido	46.6	39.9	27.4
Objeto C	Real	45	35	20
	Medido	46.7	32.4	16.3
Objeto D	Real	70	22	20
	Medido	69.9	18.9	14.6

O referencial escolhido, na câmara *RGB*, para este teste está representado na Figura 5.5.

Figura 5.5: Calibração extrínseca da câmara *RGB*, numa terceira posição

Os resultados dos parâmetros extrínsecos de calibração resultam no vetor de translação T_{RGB} e na matriz de rotação R_{RGB} que se segue.

$$T_{RGB} = \begin{bmatrix} -258.62 \\ 29.97 \\ 1275.79 \end{bmatrix}, \quad R_{RGB} = \begin{bmatrix} 0.936612 & 0.005602 & -0.350324 \\ 0.164090 & -0.890452 & 0.424464 \\ -0.309569 & -0.455043 & -0.834927 \end{bmatrix}$$

A Tabela 5.6, contém as medidas obtidas no referencial do mundo da câmara *RGB*.

Tabela 5.6: Terceiro teste de medidas do mundo real da câmara da RGB

Câmara Colorida		X	Y	Z
Objeto A	Real	20	22	20
	Medido	19.2	18.1	16
Objeto B	Real	45	35	30
	Medido	46.8	32.3	26.7
Objeto C	Real	45	35	20
	Medido	47.2	31.5	16.0
Objeto D	Real	70	22	20
	Medido	70.7	18.5	14.2

Ao observar as tabelas respetivas a este terceiro teste conclui-se que os valores medidos no eixo Y e Z sofrem uma variação considerável. Desta forma podemos concluir que dos três ensaios realizados a melhor forma de retirar medidas com alguma precisão será com o *Kinect* na primeira posição, pois os eixos X e Y são bastante próximos. Embora o eixo do Z seja mais difícil de medir quando existem objetos nas extremidades da imagem, no centro da mesma a medida de Z já é mais próxima do valor verdadeiro.

5.3 Aquisição e processamento de sinal

O Arduino nano será responsável por processar o sinal recebido dos sensores carga, do MPU6050 e do potenciômetro. Para que o sensor de força envie o sinal para o Arduino este é amplificado por um conversor analógico para digital de 24 bits, modelo HX711, sendo que o sinal é enviado através dos pinos DT e CLK. O MPU6050 envia o sinal pelas portas SDA e SCL. O potenciômetro é ligado diretamente à porta analógica, sem ser necessário qualquer tipo de tratamento do sinal.

5.3.1 Calibração do HX711 para sensores de força

O HX711 é um conversor analógico para digital de 24 bits de precisão e é maioritariamente utilizado para amplificar e converter sinais recebidos de sensores que funcionam com o princípio da ponte de wheatstone, sendo assim o ADC mais apropriado para o sensor de força escolhido. No Arduino o sinal recebido do HX711 é processado recorrendo à biblioteca *HX711-master* [37]. Quando os sensores estão ativos esta biblioteca mostra o valor da massa de um objeto. Para que este valor de massa seja o correto, faz-se a calibração dos sensores, utilizando um objeto de massa 64 gramas, que foi pesado numa balança de laboratório.

De seguida fomos verificar se a calibração foi bem sucedida. Para tal recolheu-se 26 amostras do objeto com massa 64 gramas e seguidamente colocou-se um outro objeto de 1000 gramas e observou-se as 26 amostras. O Arduino ao fim de um segundo recebeu um total de 13 amostras lidos do HX711, logo o valor de 26 amostras corresponde a um conjunto de leituras ao fim de dois segundos. Os resultados destas amostras podem ser observados na Tabela 5.7

Tabela 5.7: Amostra de valores de massa de dois objetos

	Objeto de 64 gramas	Objeto de 1000 gramas
Massa medida	63	999
	63	998
	63	998
	63	997
	63	998
	63	998
	63	998
	64	998
	64	997
	64	998
	64	998
	63	998
	63	998
	64	997
	63	998
	63	998
	64	998
	64	998
	63	997
	63	998
	64	998
	63	1005
	63	998
	63	998
	64	999
	64	999
	Média	63,4
	Desvio padrão	0,5
		998,2
		1,4

Ao observar a Tabela 2.1 verifica-se que há algumas variações nas medidas obtidas em relação ao verdadeiro peso dos objetos. Calculando, a partir destas amostras o valor da média de cada um dos objetos conclui-se que o valor médio do objeto de 64 gramas é de 63.38 gramas e para o objeto de 1000 gramas é de 998.23 gramas. O desvio padrão do peso dos objetos é respetivamente 0.49 e 1.45. Desta forma podemos concluir que os sensores estão bem calibrados, já que em relação ao valor verdadeiro dos objetos não existe uma grande variação.

Estes valores consistem em valores de calibração que medem a massa do objeto. O que é pretendido é a medida da força, em unidades SI (Newton), com que o objeto é agarrado. Para conseguir esta medida de força multiplica-se o valor de massa em unidades SI (quilograma) pela aceleração gravítica 9,8 metros por segundo. A Figura 5.6 mostra um gráfico com os valores de força. As setas a vermelho indicam o agarrar de um copo de água plástico vazio, de seguida o mesmo copo com água no seu interior, uma garrafa de água de 330 mililitros e por fim novamente

o copo de água cheio.

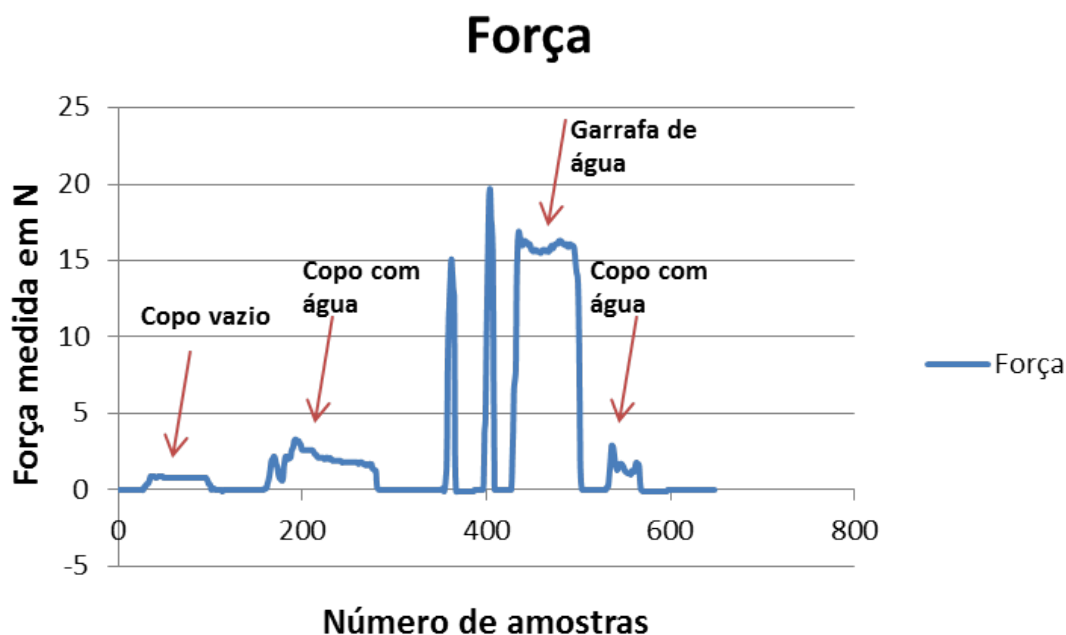


Figura 5.6: Força de medida pelos sensores

Ao agarrar o copo vazio exerceu uma força média de 0,8N com ângulo e abertura de 63 graus. Quando o copo foi preenchido com água já foi necessária uma força média de 2,1N, com um ângulo de abertura de 31 graus. A força média necessária para agarrar a garrafa foi de 15,8N com um ângulo de 50 graus. Este ângulo é maior do que no copo, visto que o plástico na garrafa é mais robusto, daí sofrer menos deformação. Por último agarrou-se novamente o copo com água, desta vez com força média de 1,8N, e com um ângulo de abertura maior 35 graus. Os dois picos de força que surgem no gráfico entre o segundo e o terceiro teste, são duas tentativas de agarrar a garrafa, mas que não foram bem sucedidas.

5.3.2 Calibração de valores do acelerómetro

As medições no MPU6050 encontram-se na Tabela 5.8. Esta exibe um conjunto de valores do acelerómetro segundo os três eixos X, Y e Z. Este valores foram obtidos mantendo o acelerómetro numa superfície horizontal, para verificar se AX e Ay tomam valor próximo de 0.

O valor de sensibilidade escolhido para a saída digital do acelerómetro do MPU6050 é $\pm 2g$, sendo que desta forma espera-se que as medidas RAW sejam próximas de 0 para os eixos X e Y e 16384 para o eixo Z [38]. Pela Tabela 5.8 é possível concluir que o MPU6050 está a funcionar corretamente.

Tabela 5.8: Valores RAW do acelerómetro

	AX	AY	AZ
Medidas acelerómetro	0	72	15252
	-20	-32	15264
	8	76	15196
	8	4	15188
	-36	20	15240
	-28	-12	15124
	-40	32	15080
	24	28	15208
	60	-4	15140
	-76	12	15032
	-64	20	15072
	-80	-64	15148
	-76	-64	15308
	88	-8	15148
	-32	20	15148
	-12	-68	15088
	-72	44	15164
	52	60	15128
	-20	24	15196
	100	20	15096
	-56	-76	15192
	20	24	15172
	-24	-40	15108
	8	-16	15172
	0	-84	15216
	36	-16	15264
Média	-8,92308	-1,07692	15167,08
Desvio padrão	48,45055	43,93783	65,89789

Após verificar que os valores medidos do acelerómetro eram corretos, testou-se agarrar um copo de plástico vazio e em seguida com água. No terceiro ensaio foi com utilizada uma garrafa de água de 330 mililitros e por fim novamente o o copo com água. Na Figura 5.7 é possível observar o gráfico de inclinação da garra. A linha a azul representa os valores do ângulo, com filtro de média, de forma a se compreender melhor a inclinação. Por sua vez, a linha vermelha diz respeito aos valores de ângulo sem filtro.

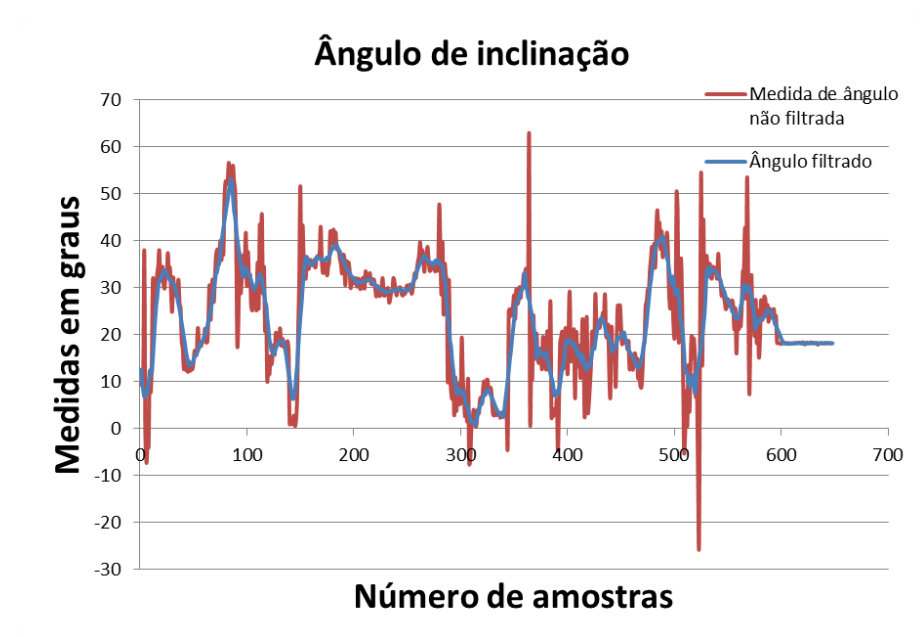


Figura 5.7: Ângulo de inclinação da garra

5.4 Resultados das medidas da garra

Aqui podemos observar os primeiros resultados obtidos das medidas do mundo da garra juntamente com o valor da restante instrumentação. A Figura 5.8 mostra o posicionamento da garra.

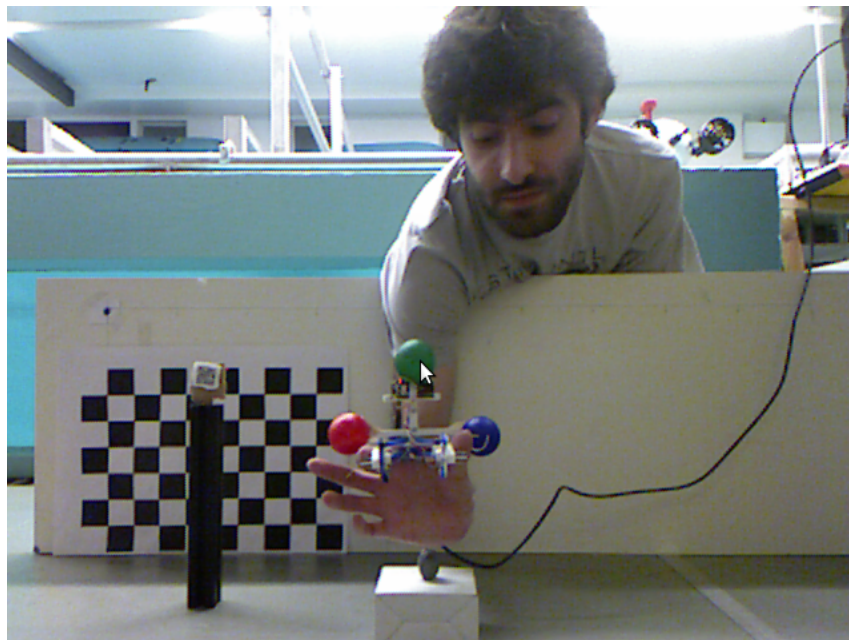


Figura 5.8: Posição da garra aberta

Nesta posição, medindo as coordenadas a partir do marcador verde, as medidas na imagem *IR*

são (44.9;22.1;30.4) centímetros e a imagem *RGB* são (44.9;22.6;30.5) centímetros. O valor da força, por sua vez, é de 0 e o valor do ângulo de abertura é de 46 graus.

A Figura 5.9 mostra o posicionamento da garra, quando esta se encontra fechada sem agarrar qualquer objeto.

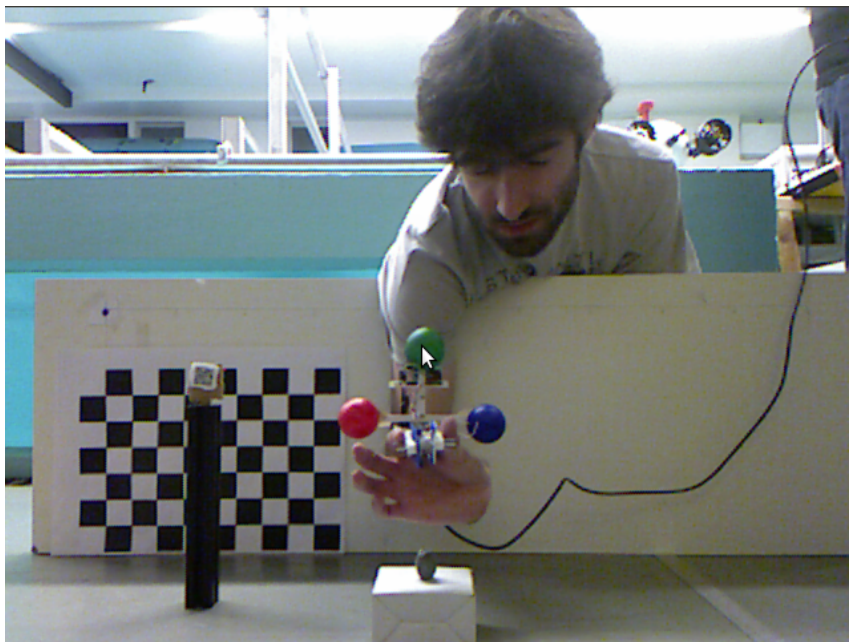


Figura 5.9: Posição da garra fechada

Nesta posição, medindo as coordenadas a partir do marcador verde, as medidas na imagem *IR* são (44.9;23.5;29.3) centímetros e a imagem *RGB* são (45.5;23.9;29.3) centímetros. O valor da força, por sua vez, é de 0 e o valor do ângulo de abertura é de 2 graus.

De seguida foram realizados três testes com todo o sistema de instrumentação incorporado. Utilizou-se a garra para mover dois objetos. O segundo objeto foi usado no terceiro ensaio, mas agarrado pela garra numa posição diferente.

O primeiro objeto a ser deslocado para outro sítio foi o objeto de 64 gramas utilizado anteriormente para calibrar os sensores de carga. Na Figura 5.10 vemos o momento em que o objeto é agarrado. De acordo com as coordenadas medidas a partir do marcador verde as medidas da câmara *IR* são (41.5;11.9;22.2) centímetros e as medidas da câmara *RGB* são (42.1;11.4;22.4) centímetros.

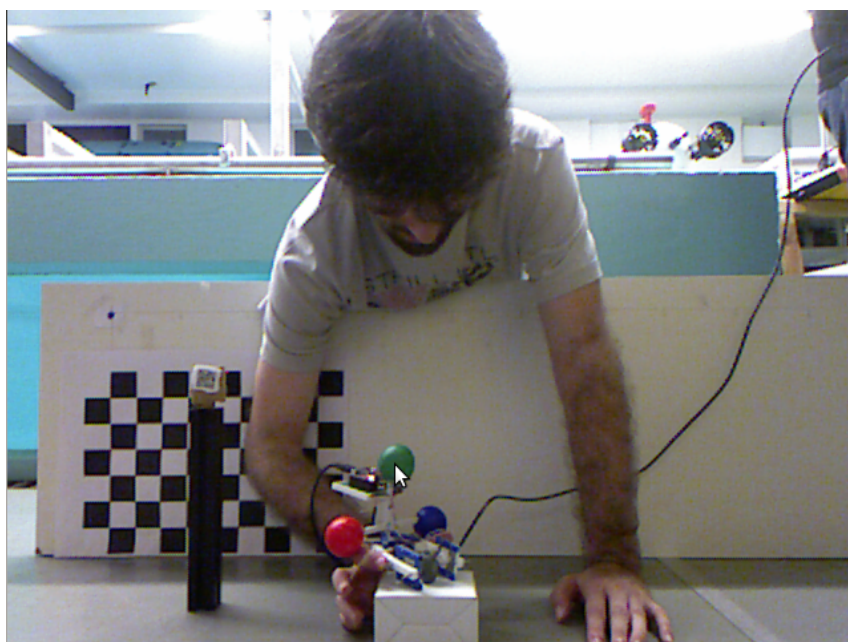


Figura 5.10: Agarrar o objeto

A Figura 5.11 mostra a posição da garra antes de agarrar o objeto. Este objeto é agarrado pelos sensores e mudado para outra posição.

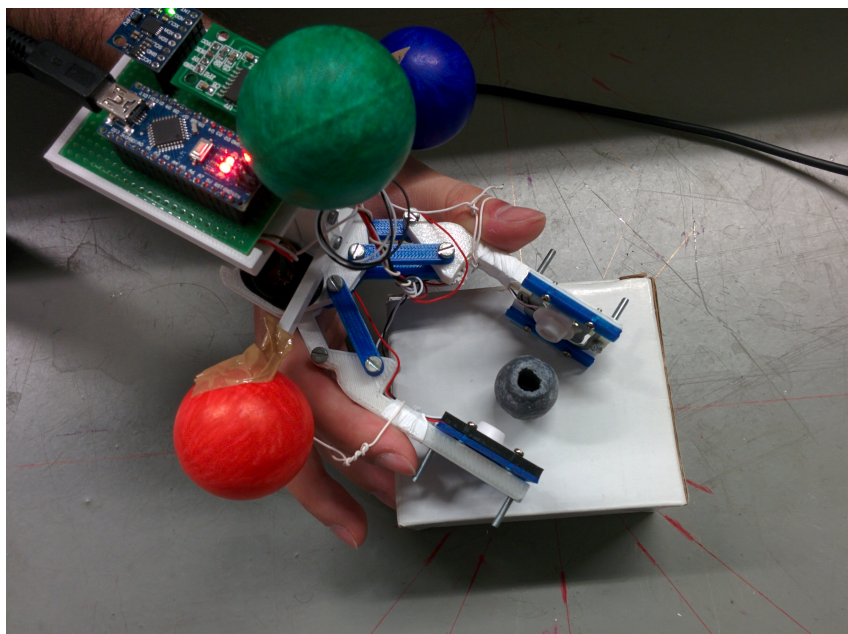


Figura 5.11: Preparação para agarrar objeto

A Figura 5.12 mostra o movimento sofrido pelo objeto. Nesta posição as coordenadas indicadas pela câmara *IR* são (43.1;26.2;29.1) centímetros e as coordenadas da câmara *RGB* são (43.8;26.2;29.0) centímetros.

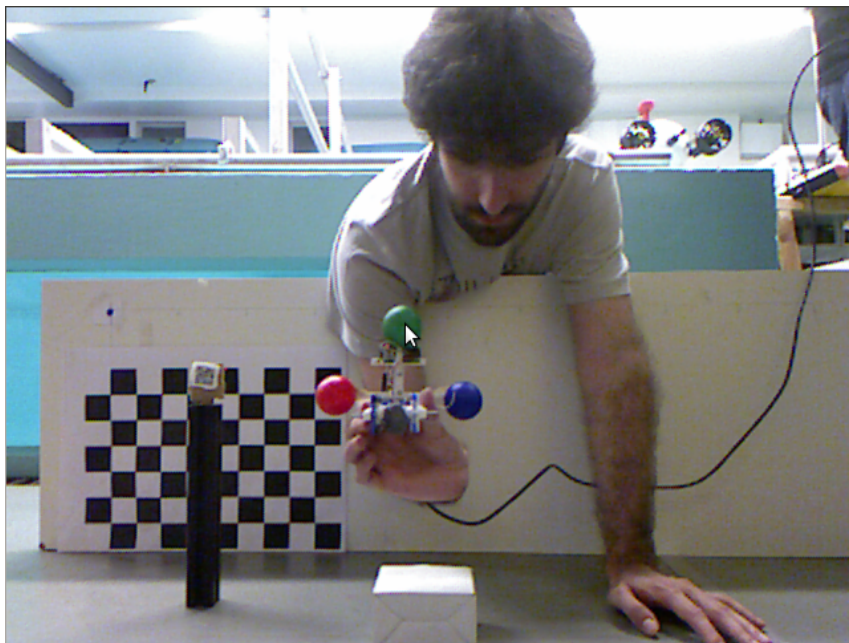


Figura 5.12: Movendo o objeto

A Figura 5.13 exibe o objeto agarrado pelos sensores e numa posição diferente da original.

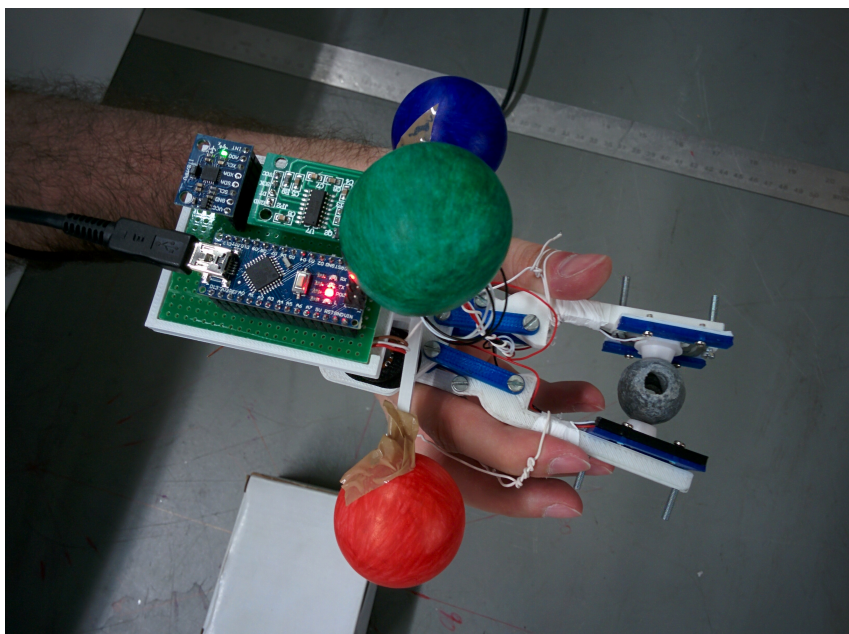


Figura 5.13: Objeto agarrado pelos sensores

A Tabela 5.9 mostra os valores medidos do acelerómetro, nos três eixos, bem como a força que o utilizador exerceu para agarrar o objeto e o ângulo de abertura.

Tabela 5.9: Medidas adquiridas da garra

Número de amostras	AX	AY	AZ	Força	Ângulo
1	2688	2870	14374	9,4	15
2	2950	3497	14394	10,7	14
3	2736	2521	14447	10,6	14
4	2755	2627	14434	10,1	14
5	2799	2657	14529	10,4	14
6	2795	2405	14499	10,6	14
7	2509	2068	14445	11,7	13
8	3031	3188	14531	12,7	13
9	3413	3682	14298	12	13
10	2556	-1001	15114	10,4	13
11	2475	1985	15216	10,1	13
12	4316	4329	15035	10,8	13
13	1647	3550	14258	11,8	13
14	1571	3672	13242	11,4	13
15	622	3140	14504	11,4	13
16	1209	5217	13923	12,1	13
17	-141	3156	14389	12,2	13
18	831	3289	13913	11,1	13
19	777	3045	14310	10,1	13
20	196	2456	14675	9,3	13
21	973	3867	14719	9	13
22	46	3138	15123	8,8	13
23	1570	3798	14685	8,6	13
24	296	3925	14536	8,5	13
25	178	6702	14724	8,4	13
26	3228	3074	13835	8	13
Média				10,4	13,3
Desvio padrão				1,3	0,5

Ao observar a Tabela 5.9 conclui-se que a força média aplicada pelo utilizador é de 10.4N e o ângulo de abertura dos dedos da garra é de 13.3 graus.

O segundo objeto a ser testado foi uma pilha AA. Esta foi colocada em posição vertical e agarrada pela garra enquanto permaneceu nessa posição. Esta foi utilizada também no terceiro ensaio, mas aqui a sua posição de agarra foi horizontal.

A Figura 5.14 mostra quando o objeto é agarrado. De acordo com as coordenadas medidas a partir do marcador verde as medidas da câmara *IR* são (42.7;11.7;21.8) centímetros e as medidas da câmara *RGB* são (43.4;11.7;21.9) centímetros.

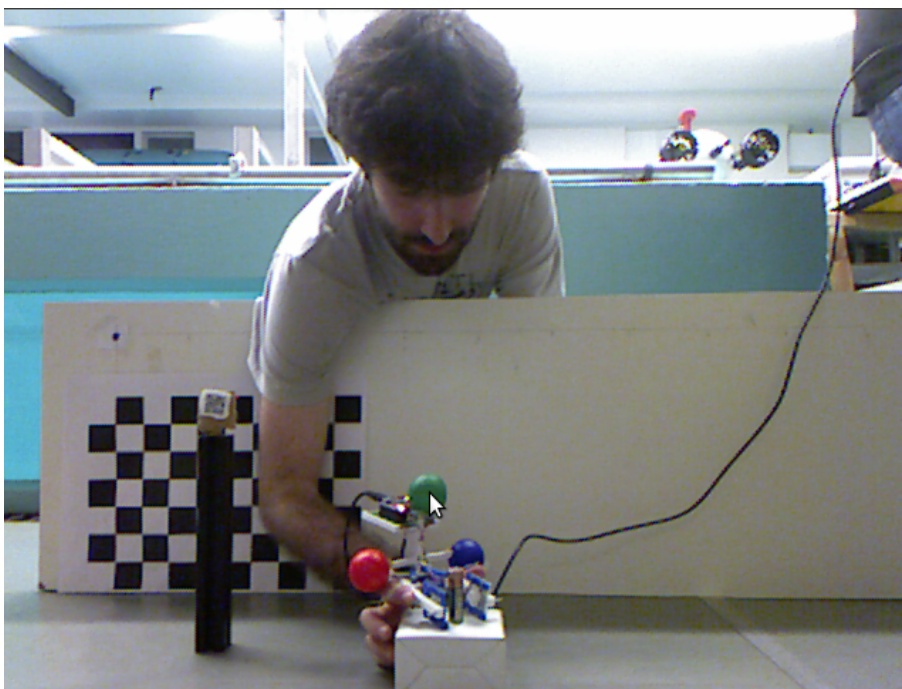


Figura 5.14: Agarrar o objeto

A Figura 5.15 mostra a garra na posição inicial, sendo que esta posteriormente agarra a pilha.

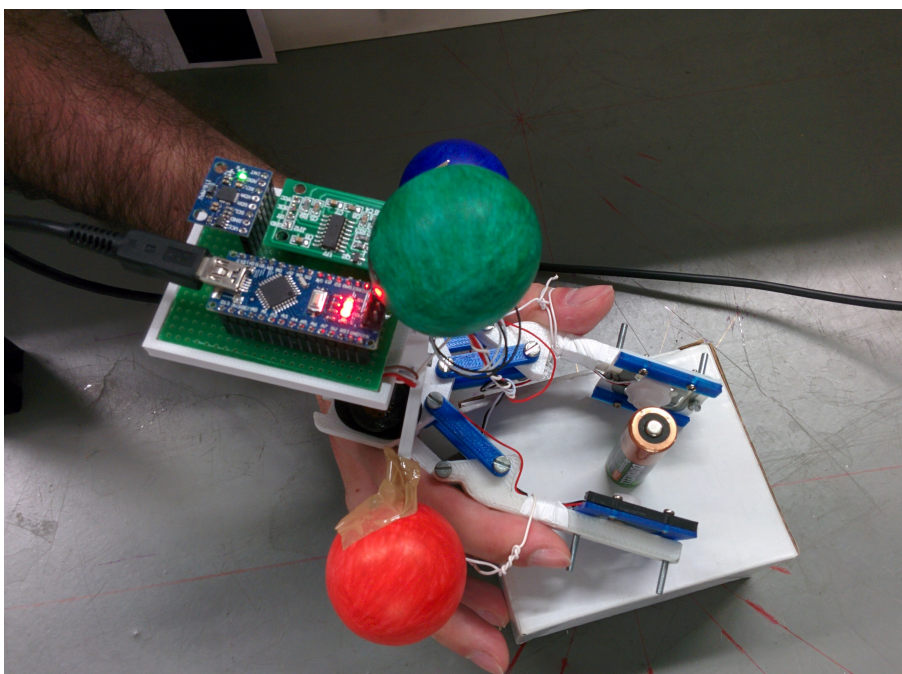


Figura 5.15: Preparação para agarrar a pilha

A Figura 5.16 mostra o movimento sofrido pelo objeto. Nesta posição as coordenadas indicadas pela câmara *IR* são (38.9;26.2;27.9) centímetros e as coordenadas da câmara *RGB* são

(39.5;26.7;27.9) centímetros.

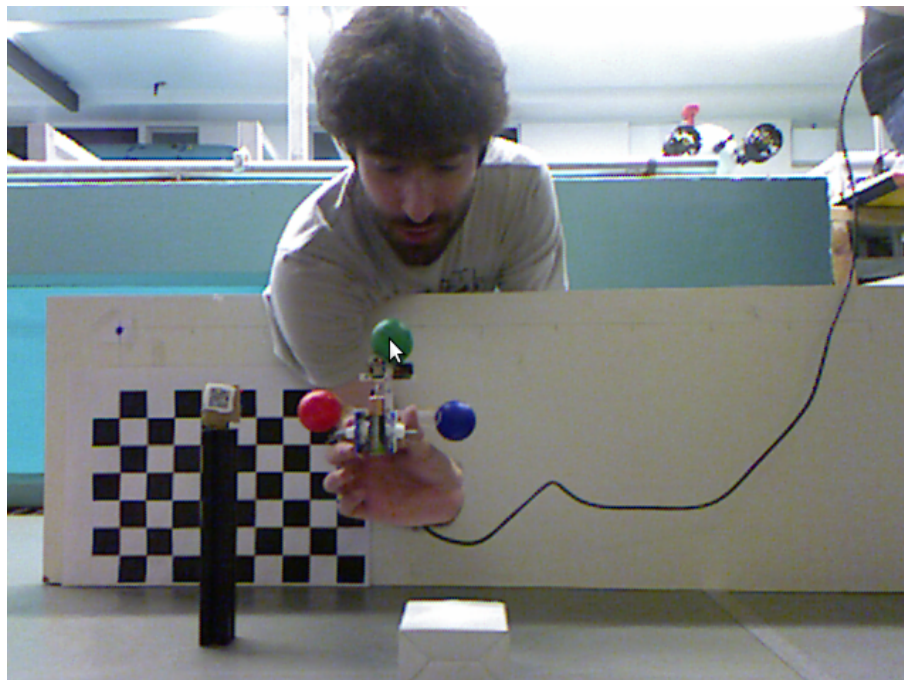


Figura 5.16: Movendo o objeto

A Figura 5.17 mostra a pilha agarrada pelos sensores. A garra transportou a pilha para a posição (39.5;26.7;27.9) centímetros.

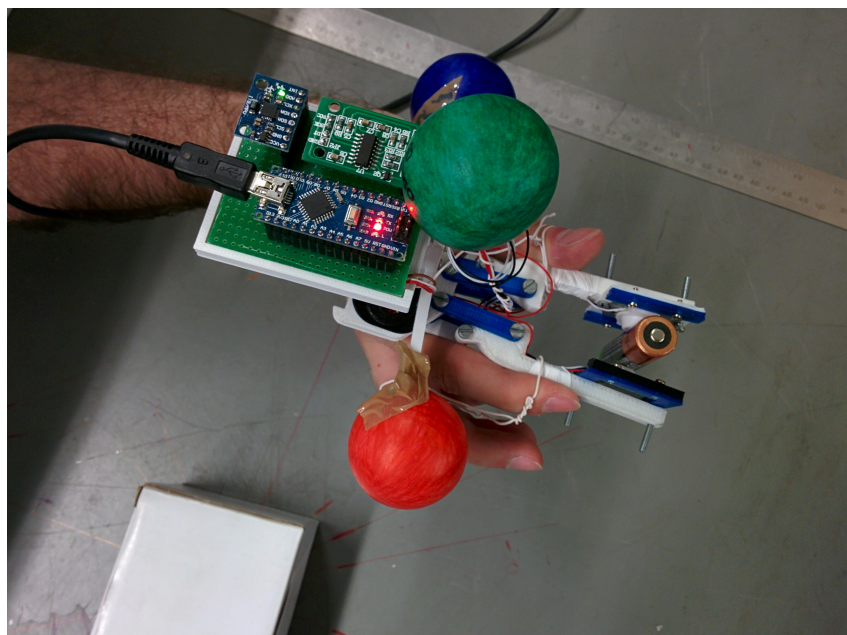


Figura 5.17: Pilha agarrada pelos sensores

Tabela 5.10: Medidas adquiridas de um segundo objeto

Número de amostras	AX	AY	AZ	Força N	Ângulo
1	2960	1894	14459	9,9	7
2	2902	1494	14541	9,9	7
3	2953	1921	14492	9,4	7
4	3081	2737	15002	9,4	7
5	3863	1352	16005	9,3	7
6	1556	1190	15475	7,9	7
7	1120	1939	15003	7,1	7
8	411	271	14339	6,3	7
9	825	1958	13220	7,1	7
10	2068	1386	14540	7,8	7
11	2257	1226	15875	7,7	7
12	1979	519	14045	7,6	7
13	2024	993	14493	7,7	7
14	2759	6974	14601	7,2	7
15	417	1504	14185	6,9	7
16	1664	3332	14840	6,5	7
17	1342	798	14849	7,3	7
18	544	1520	14742	8,7	7
19	2476	836	15790	8,9	7
20	1247	2427	14596	8,9	7
21	1576	1881	14732	9,4	7
22	2464	2451	15293	9,5	7
23	1612	2722	14557	9,1	7
24	2197	2790	14907	9,3	7
25	1459	2891	14515	9,1	6
26	1979	-379	14326	8,4	7
Média				8,3	6,9
Desvio padrão				1,1	0,2

Ao observar a Tabela 5.10 conclui-se que a pilha AA usada para teste tem espessura inferior ao primeiro objeto, visto que ao agarrá-la o ângulo de abertura é inferior, sendo a média deste valor de 6.9 graus. Pode-se ainda verificar que o utilizador utilizou em média 8,3N de força para agarrar o objeto, sendo esta força inferior à que foi aplicada no objeto do primeiro ensaio.

A Figura 5.18 exibe a pilha AA a ser agarrada numa posição horizontal. Nesta posição as coordenadas indicadas pela câmara *IR* são (31.2;12.5;31.7) centímetros e as coordenadas da câmara *RGB* são (31.1;12.2;31.9) centímetros.



Figura 5.18: Objeto em nova posição

A Figura 5.19 mostra a pilha agarrada pelos sensores, numa posição horizontal.

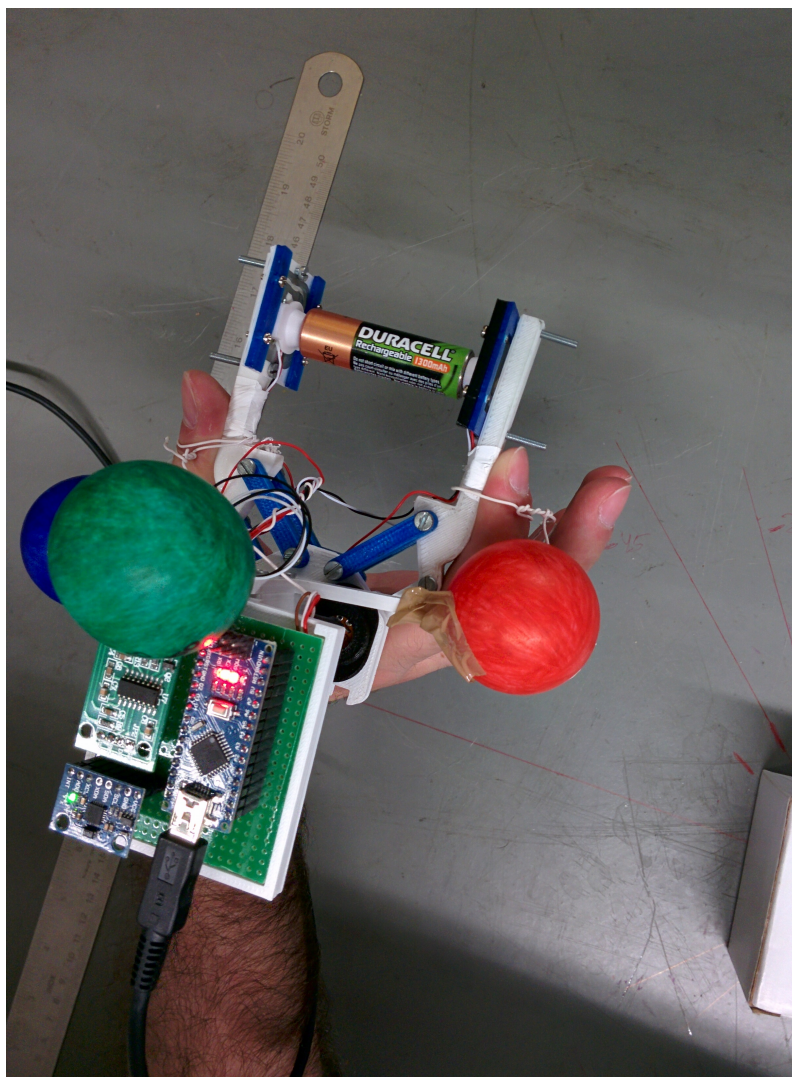


Figura 5.19: Pilha agarrada pelos sensores

A Tabela 5.11 mostra os valores medidos ao agarrar a pilha numa posição diferente. Conclui-se que a força aplicada para agarrar a mesma não sofre grandes alterações em relação ao segundo ensaio, tendo um valor médio de 7,7N. Já a abertura da garra aumenta para um ângulo de abertura de 61 graus, com a particularidade de este ângulo ser constante, não existindo qualquer desvio no valor lido.

Tabela 5.11: Medidas da garra com pilha AA em posição horizontal

Número de amostras	AX	AY	AZ	Força N	Ângulo
1	1813	297	14873	7,1	61
2	1375	-1182	14753	7,2	61
3	1322	57	14924	7,2	61
4	1571	-689	14875	7,2	61
5	1666	-700	14729	7,2	61
6	1361	-1541	14742	7,6	61
7	1852	-428	14726	8,4	61
8	1838	-865	14816	7,9	61
9	1863	-655	14930	7,9	61
10	1841	-770	14882	7,8	61
11	1883	-294	14779	7,8	61
12	1940	-997	14756	7,8	61
13	1926	-961	14743	7,8	61
14	2057	-162	14803	7,8	61
15	2249	-476	14656	7,8	61
16	2152	-858	14557	7,8	61
17	2087	-725	14747	7,7	61
18	2355	-222	14877	7,8	61
19	2262	-686	14728	7,8	61
20	2146	-1071	14644	7,8	61
21	2185	-506	14825	7,8	61
22	2160	-692	14771	7,8	61
23	2490	-35	14923	7,8	61
24	1840	-396	14600	7,7	61
25	2343	-634	14532	7,7	61
26	1802	-1705	14833	7,7	61
Média				7,688462	61
Desvio padrão				0,281926	0

5.5 Conclusões

Neste capítulo verificou-se que todos os módulos de hardware funcionam devidamente. Após a calibração destes procedeu-se a uma série de testes, já descritos que validaram o sistema desenvolvido. Verificou-se as medidas das coordenadas no espaço 3D por parte do *Kinect*, a orientação da garra, a força exercida ao agarrar vários objetos e o ângulo de abertura da garra. No capítulo seguinte serão descritas as conclusões finais de todo o projeto.

Capítulo 6

Conclusão e Trabalho Futuro

No presente capítulo serão descritas as principais conclusões relativas às ferramentas utilizadas na concretização da dissertação conjugadas com o estado de desenvolvimento da mesma. Igualmente, serão ainda tecidas algumas considerações sobre possíveis implementações futuras tendo em vista o aperfeiçoamento do projeto.

6.1 Conclusões

O principal objetivo desta dissertação consistia no desenvolvimento de uma aplicação robótica tendo por base a técnica de aprendizagem por demonstração. Neste sentido, o método baseava-se na recolha de informação através de vários sensores, do movimento de uma ferramenta robótica desenhada para deslocar objetos.

Numa fase inicial, o foco do trabalho incidiu na visão *3D* com recurso ao sensor *Kinect*. Por meio desta abordagem foi possível identificar a garra no espaço *3D* e descobrir a sua posição em coordenadas cartesianas com bons resultados. O procedimento adotado baseava-se na deteção de marcadores de cor que foram adicionados auxiliando a identificação do *grripper*.

Foi criada em *SolidWorks* e posteriormente impressa uma garra capaz de auxiliar na aprendizagem por demonstração. Este protótipo e a consequente, implementação na prática, permitiu adicionar sensores para medir uma série de parâmetros importantes. Os parâmetros medidos por esse sensores foram: força, orientação e ângulo. A força foi medida através de dois sensores de carga fixados na ponta dos dedos da garra. Estes sensores funcionam, em conjunto, com base na ponte de wheatstone. Foi utilizado um acelerómetro de três eixos que consegue estimar a orientação. O ângulo de abertura da garra foi obtido com recurso a um potenciómetro, convertendo tensão em ângulo.

6.2 Trabalho Futuro

Para um desenvolvimento futuro, seria interessante melhorar o método de calibração extrínseca. Assim o cálculo das coordenadas no mundo real seria mais preciso. Uma possível vantagem

desta melhoria seria remover os marcadores de cor colocados na garra de demonstração.

Quanto à garra de demonstração, poderá ser melhorada a zona de contacto entre os sensores e os objetos. Isto faria com que fosse mais seguro agarrar objetos com superfícies escorregadias.

Utilizar garras de outros tipos é uma função que futuramente poderá ser explorada. A título de exemplo, utilizar garras de três ou quatro dedos. Outra funcionalidade a ser implementada poderá ser introduzir um maior número de garras, para assim se poder ensinar dois manipuladores.

Anexo A

Anexos

A.1 Circuito Eagle do Hardware

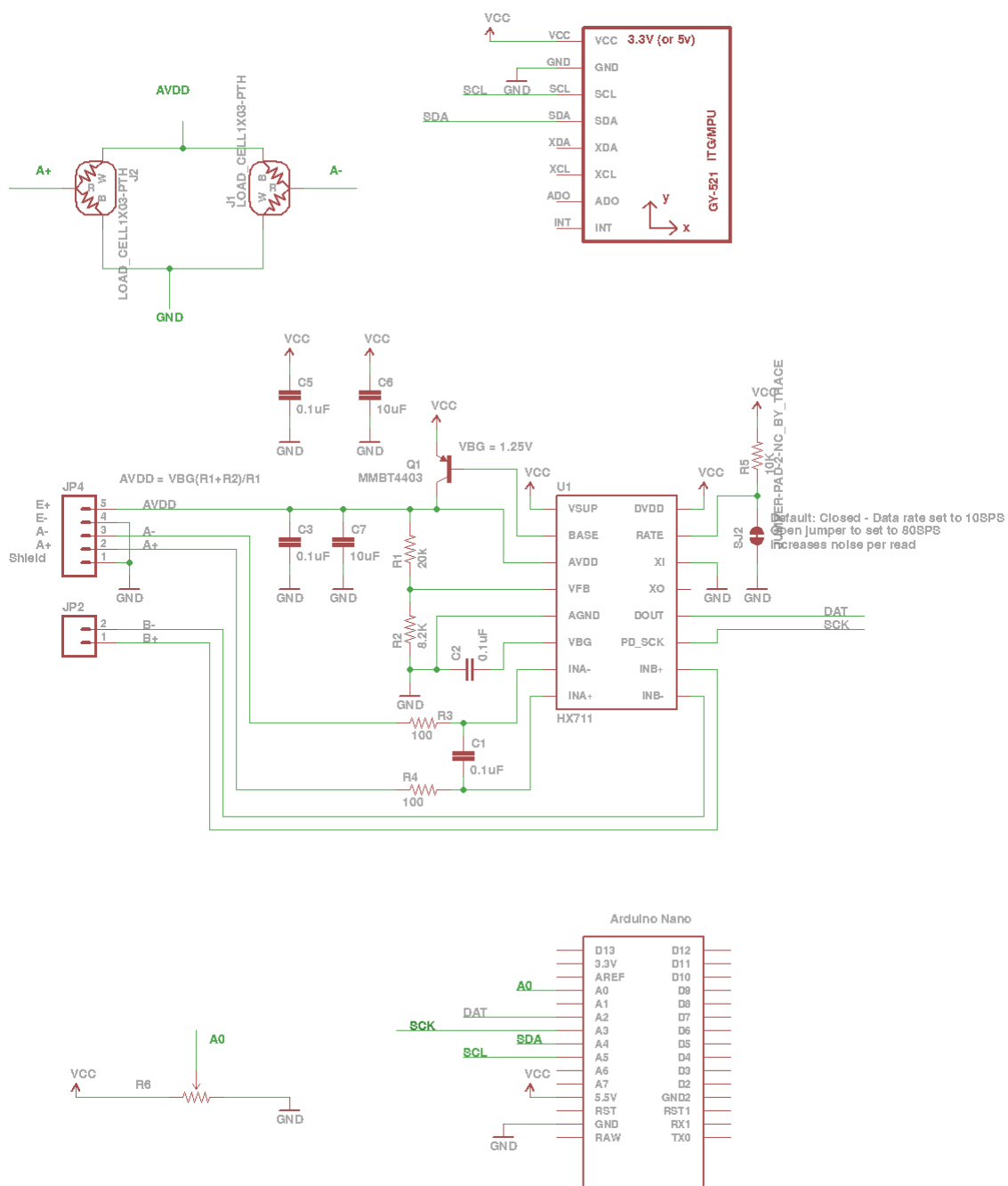
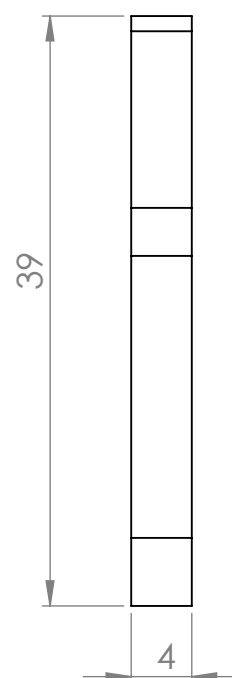
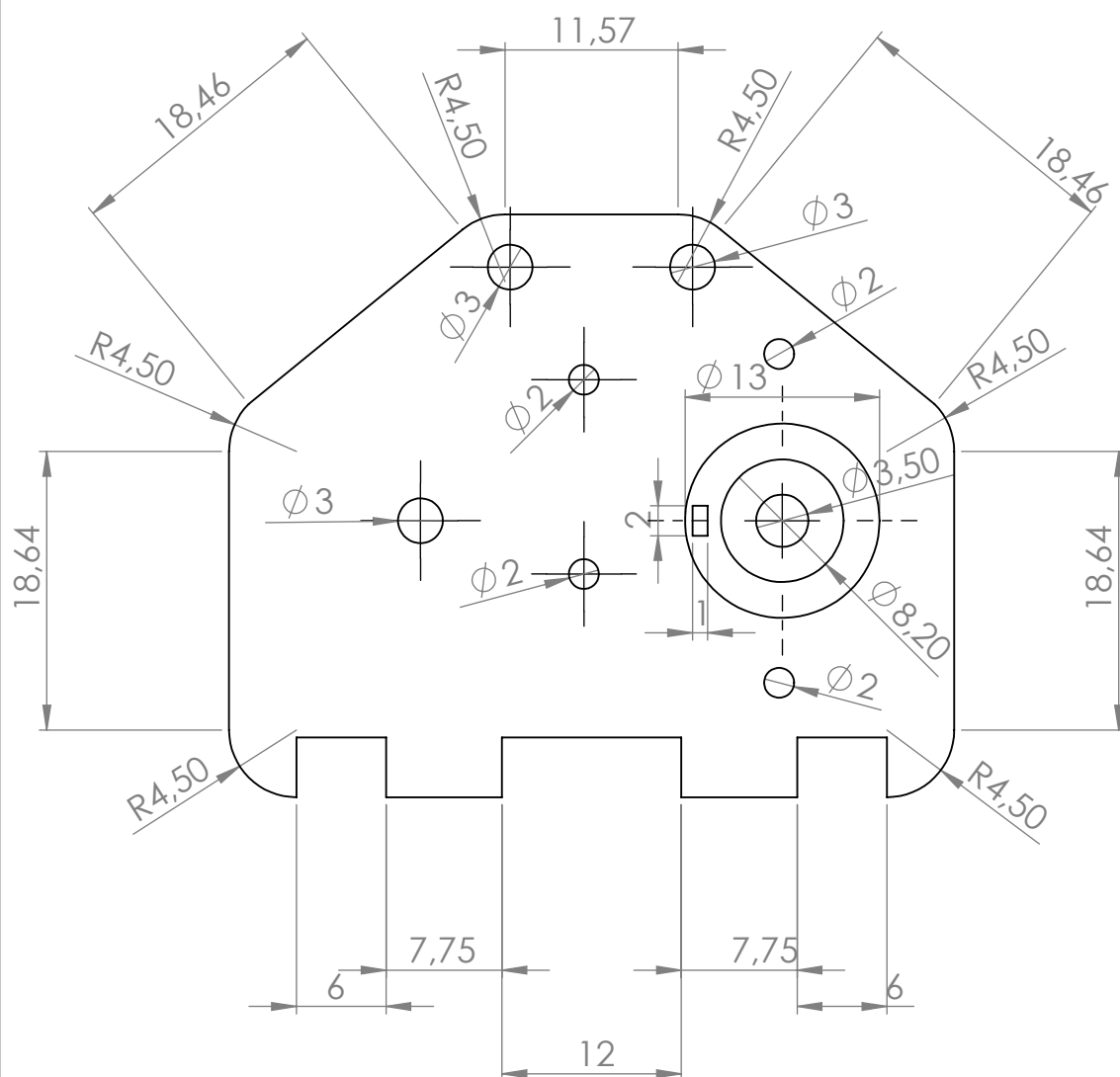
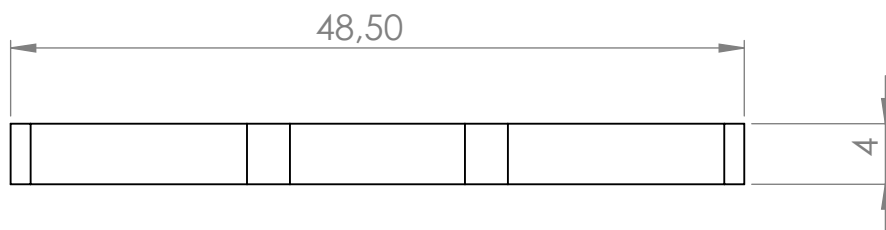


Figura A.1: Circuito Eagle do Hardware

A.2 Peças modificadas do *Mini servo gripper Plate*



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Ricardo Borges		13/06/2016
CHK'D			
APPV'D	Paulo Costa		
MFG	FEUP		
Q.A			

MATERIAL:

WEIGHT:

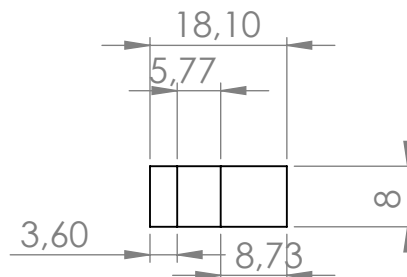
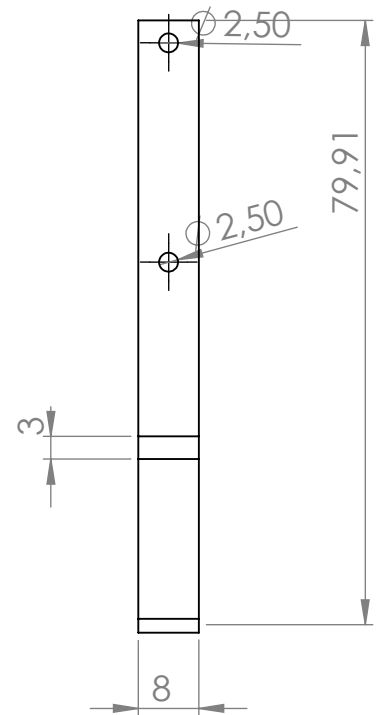
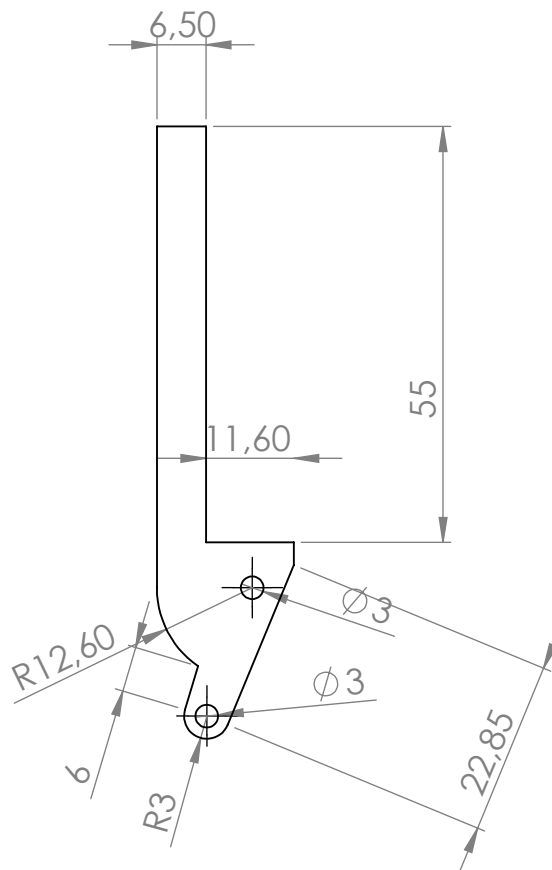
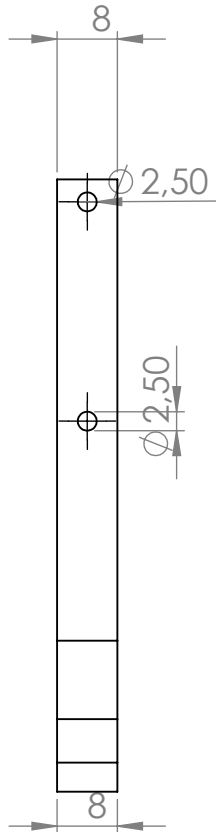
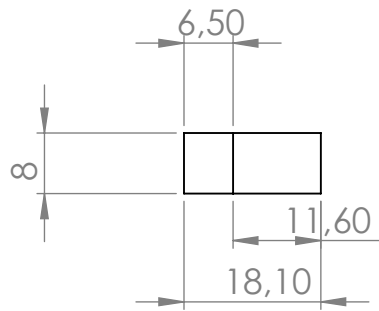
TITLE:

DWG. NO.

Gripperservoplate_rivise

SCALE:2:1

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE		
DRAWN	Ricardo Borges		26/04/2016		
CHK'D					
APPV'D	Paulo Costa				
MFG	FEUP				
Q.A				MATERIAL:	
				WEIGHT:	

TITLE:

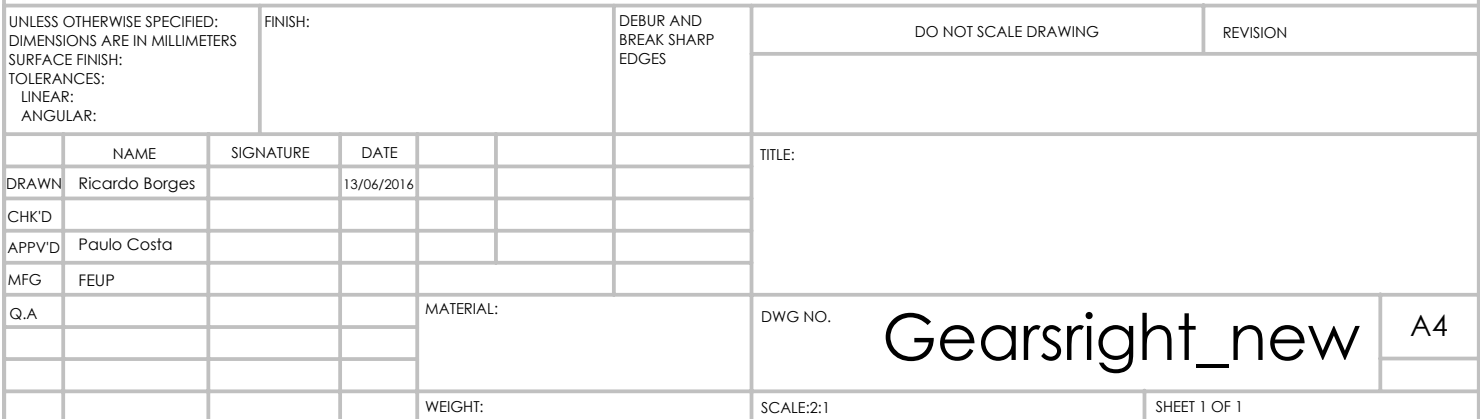
DWG NO.

SCALE:1:1

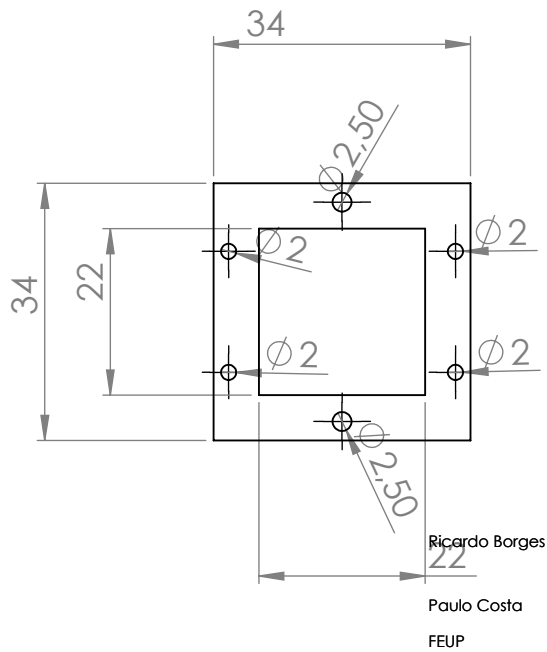
Grippart1

SHEET 1 OF 1

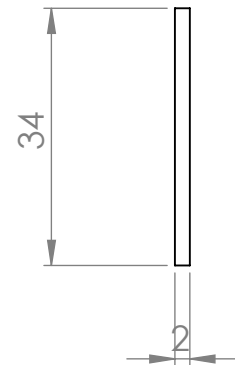
A4



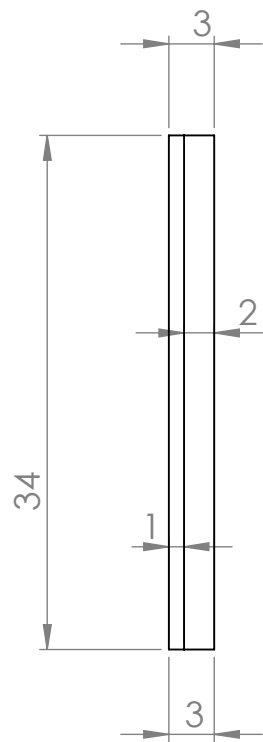
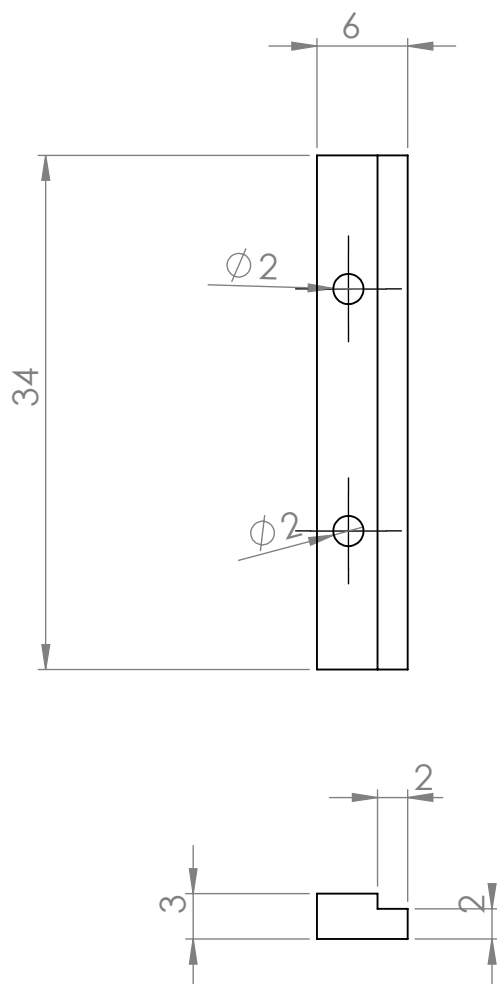
A.3 Novas peças desenvolvidas



26/04/2016

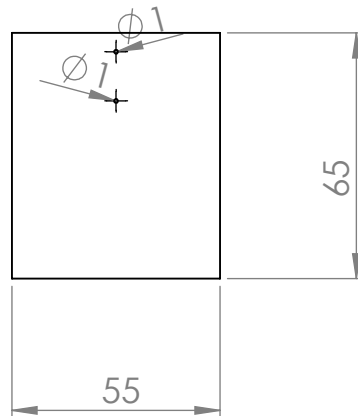
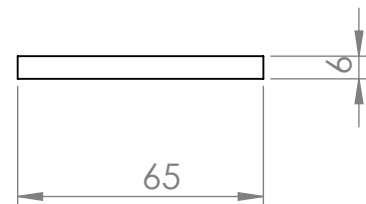
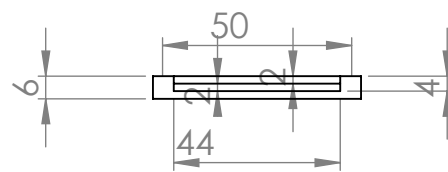
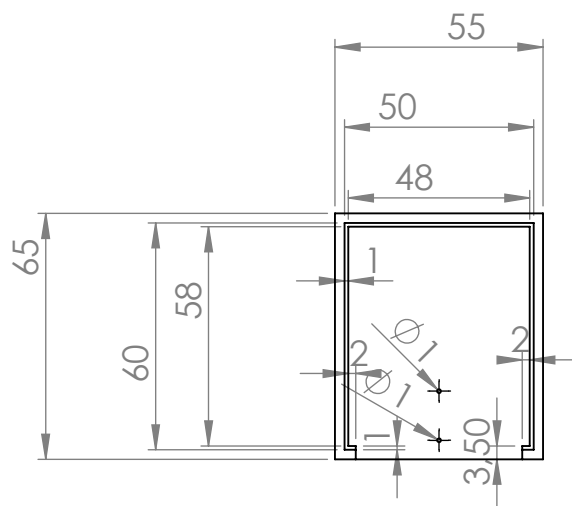


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBUR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION		
	NAME	SIGNATURE	DATE			TITLE:				
DRAWN	Ricardo Borges		15/04/2016							
CHK'D										
APPV'D	Paulo Costa									
MFG	FEUP									
Q.A				MATERIAL:		DWG NO.		sensor1		A4
				WEIGHT:		SCALE:1:1		SHEET 1 OF 1		

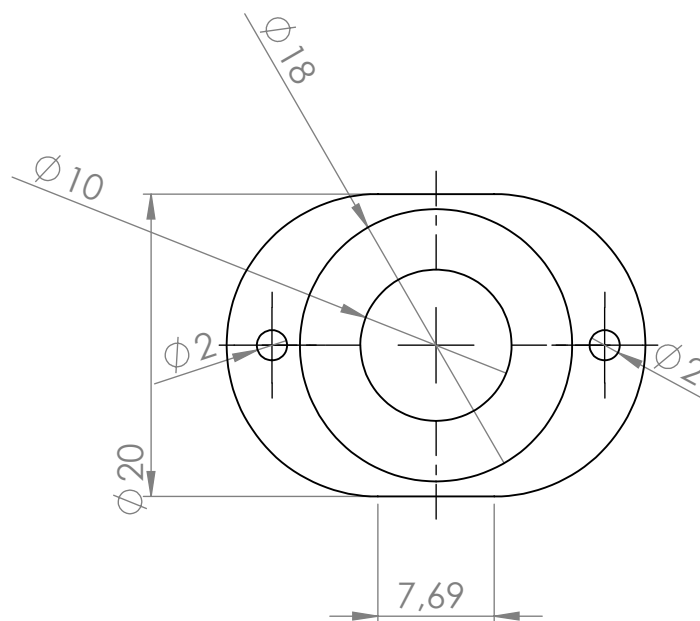
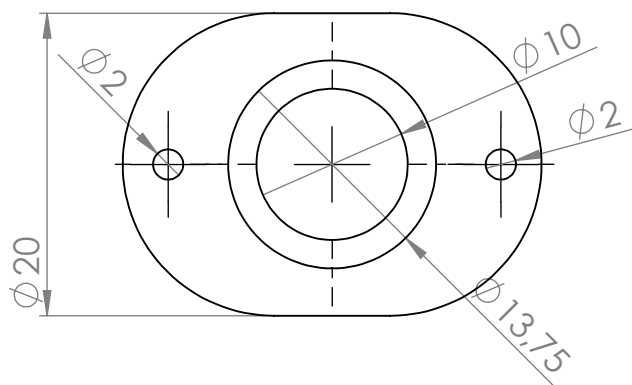
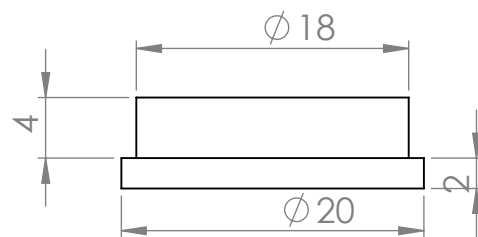
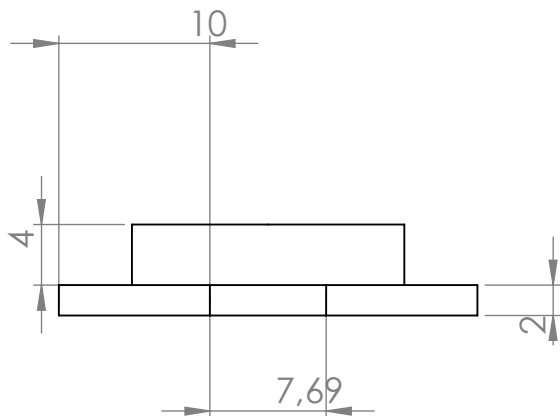


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:				FINISH:		DEBUR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION	
NAME		SIGNATURE		DATE				TITLE:			
DRAWN Ricardo Borges				15/04/2016							
CHK'D											
APPV'D Paulo Costa											
MFG FEUP											
Q.A						MATERIAL:		DWG NO.		A4	
						WEIGHT:		SCALE:2:1		SHEET 1 OF 1	

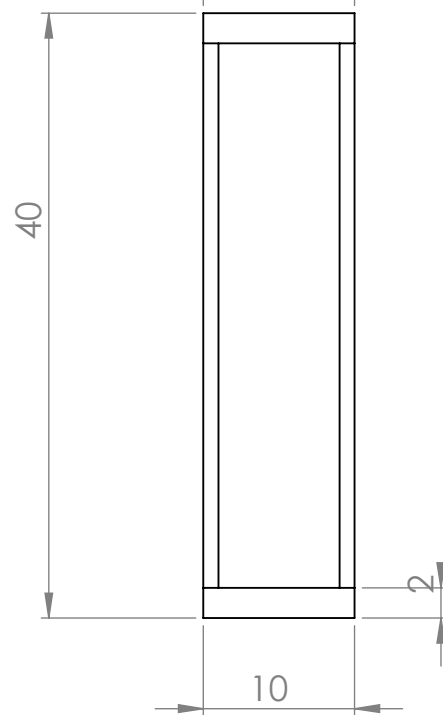
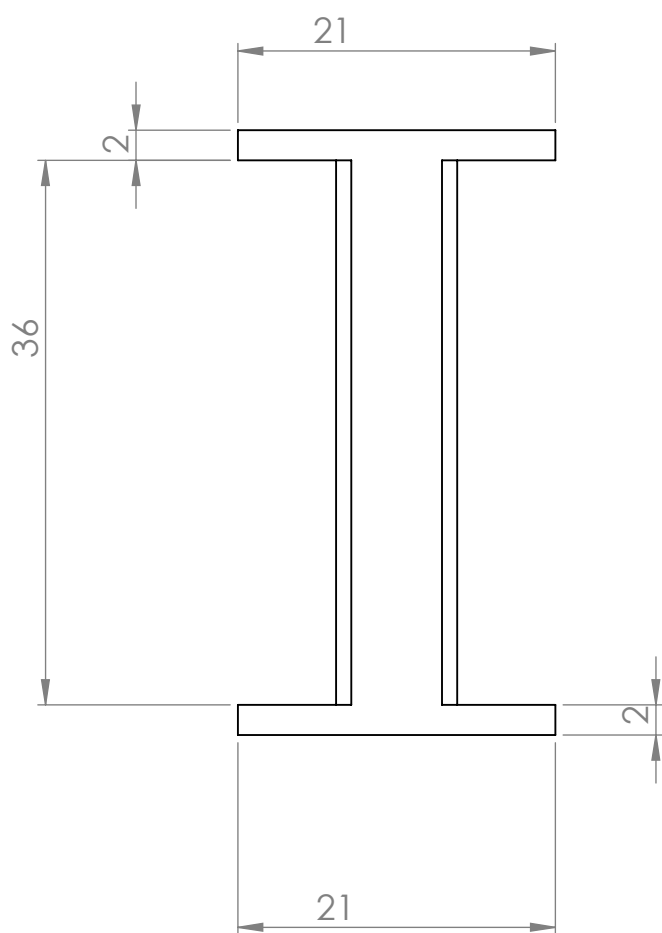
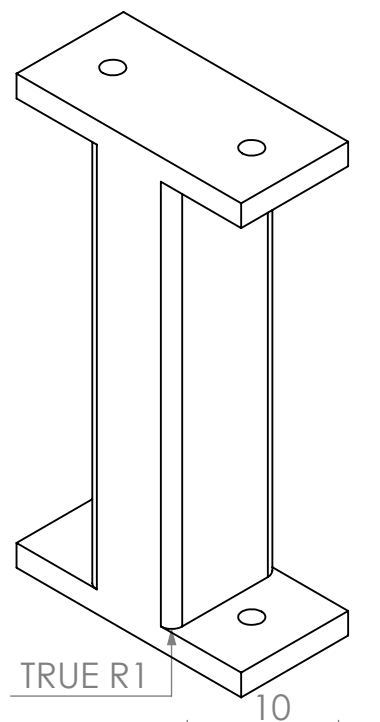
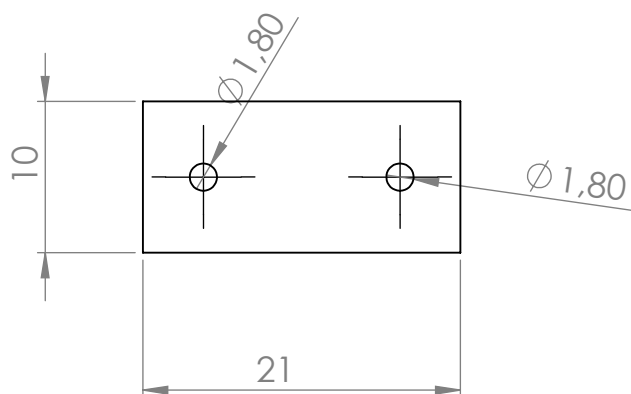
sensor2



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:				FINISH:		DEBUR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION	
NAME		SIGNATURE		DATE				TITLE:			
DRAWN Ricardo Borges				10/05/2016							
CHK'D											
APPV'D Paulo Costa											
MFG FEUP											
Q.A						MATERIAL:		DWG NO.		caixahardware	
										A4	
						WEIGHT:		SCALE:1:2		SHEET 1 OF 1	



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:				FINISH:		DEBUR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION			
								TITLE:					
DRAWN				Ricardo Borges		DATE						16/06/2016	
CHK'D													
APPV'D				Paulo Costa									
MFG				FEUP									
Q.A						MATERIAL:		DWG NO.					
								calhaPotenciometro					
								SCALE:2:1					
						WEIGHT:		SHEET 1 OF 1					



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Ricardo Borges		31/05/2016
CHK'D			
APPV'D	Paulo Costa		
MFG	FEUP		
Q.A			

MATERIAL:

TITLE:

DWG NO.

WEIGHT:

SCALE:2:1

SHEET 1 OF 1

suporte

A4

Bibliografia

- [1] “synopsys-android-virtual-prototyping-part-3-fig-12- @ m.eet.com.” [Online]. Available: <http://m.eet.com/media/1174353/synopsys-android-virtual-prototyping-part-3-fig-12-.jpg>
- [2] “kinect_aberto3 @ www.infomaniaco.com.br.” [Online]. Available: <http://www.infomaniaco.com.br/wp-content/uploads/2010/11/kinect{ }aberto3.jpg>
- [3] “Kinect-640x353 @ www.extremetech.com.” [Online]. Available: <http://www.extremetech.com/wp-content/uploads/2013/09/Kinect-640x353.png>
- [4] “01 @ dlcdnimgs.asus.com.” [Online]. Available: <https://dlcdnimgs.asus.com/20160129{ }cosmo/cosmo/websites/global/products/hahEFPMWY9UVDL7z/01.jpg>
- [5] N. Mendes, P. Neto, M. Safeea, and A. P. Moreira, *Robot 2015: Second Iberian Robotics Conference*, ser. Advances in Intelligent Systems and Computing, L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Muñoz-Martinez, Eds. Cham: Springer International Publishing, 2016, vol. 418. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84952312796{ }partnerID=tZOTx3y1>
- [6] “11021-01a-600x600 @ www.techmount.com.br.” [Online]. Available: <http://www.techmount.com.br/image/cache/data/Fotos/ArduinoUnoR3Original/11021-01a-600x600.jpg>
- [7] “3-pinhole-camera-geometry @ prateekvjoshi.files.wordpress.com.” [Online]. Available: <https://prateekvjoshi.files.wordpress.com/2014/05/3-pinhole-camera-geometry.png>
- [8] “34182443_OVR @ cnet4.cbsistatic.com.” [Online]. Available: <https://cnet4.cbsistatic.com/hub/i/r/2010/11/02/d615285f-67c3-11e3-a665-14feb5ca9861/thumbnail/770x433/053735ea7bcc9481b707f60ed7afcf3a/34182443{ }OVR.png>
- [9] “connect-hx711-to-a-three-wire-load-cell @ electronics.stackexchange.com.” [Online]. Available: <http://electronics.stackexchange.com/questions/199487/connect-hx711-to-a-three-wire-load-cell>
- [10] “hx711 @ raw.githubusercontent.com.” [Online]. Available: <https://raw.githubusercontent.com/intel-iot-devkit/upm/master/docs/images/hx711.jpeg>

- [21] K. O. Rodriguez and G. C. Chavez, “Finger Spelling Recognition from RGB-D Information Using Kernel Descriptor,” in *2013 XXVI Conference on Graphics, Patterns and Images*. IEEE, aug 2013, pp. 1–7. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84891538211&partnerID=tZOtx3y1>
- [22] T. Maps and K. Networks, “Self Organizing Maps : Fundamentals What is a Self Organizing Map ?” *Neural Networks*, pp. 1–15, 2004. [Online]. Available: <http://www.cs.bham.ac.uk/~jxb/NN/116.pdf>
- [23] Analog Devices;, “i MEMS ® Accelerometer ADXL330,” 2007. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL330.pdf>
- [24] “9375 @ www.sparkfun.com.” [Online]. Available: <https://www.sparkfun.com/products/9375>
- [25] “13330 @ www.sparkfun.com.” [Online]. Available: <https://www.sparkfun.com/products/13330>
- [26] “10245 @ www.sparkfun.com.” [Online]. Available: <https://www.sparkfun.com/products/10245>
- [27] J. Heikkilä and O. Silvén, “A Four-step Camera Calibration Procedure with Implicit Image Correction.” *Cvpr*, pp. 1106–1112, 1997. [Online]. Available: <http://dx.doi.org/10.1109/CVPR.1997.609468>
- [28] D. C. Brown, “Close-range camera calibration,” pp. 855–866, 1971. [Online]. Available: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.6358>
- [29] “parameters @ www.vision.caltech.edu.” [Online]. Available: [http://www.vision.caltech.edu/bouguetj/calib\[_\]doc/htmls/parameters.html](http://www.vision.caltech.edu/bouguetj/calib[_]doc/htmls/parameters.html)
- [30] “Caaf442E6Eb01Dd0a160Fc5Fa58Ac0E1F6229Cd8 @ Www.Vision.Caltech.Edu.” [Online]. Available: [http://www.vision.caltech.edu/bouguetj/calib\[_\]doc/](http://www.vision.caltech.edu/bouguetj/calib[_]doc/)
- [31] “Imaging_Information @ openkinect.org.” [Online]. Available: [https://openkinect.org/wiki/Imaging\[_\]Information](https://openkinect.org/wiki/Imaging[_]Information)
- [32] “thing:7109 @ www.thingiverse.com.” [Online]. Available: <http://www.thingiverse.com/thing:7109>
- [33] “B00ZC1H2PI @ www.amazon.com.” [Online]. Available: <https://www.amazon.com/BQLZR-YZC-161B-Sensor-Weighing-Indenter/dp/B00ZC1H2PI>
- [34] “getting-started-with-load-cells @ learn.sparkfun.com.” [Online]. Available: <https://learn.sparkfun.com/tutorials/getting-started-with-load-cells>

- [35] “libfreenect @ github.com.” [Online]. Available: <https://github.com/OpenKinect/libfreenect>
- [36] “5dpo @ wiki.freepascal.org.” [Online]. Available: <http://wiki.freepascal.org/5dpo>
- [37] “HX711 @ github.com.” [Online]. Available: <https://github.com/bogde/HX711>
- [38] “12e9e9bbb7cdc2fe841130cda20a39a40b6d64a9 @ www.i2cdevlib.com.” [Online]. Available: <http://www.i2cdevlib.com/forums/topic/4-understanding-raw-values-of-accelerometer-and-gyrometer/>